MOBILE MANIPULATION IN UNSTRUCTURED ENVIRONMENTS WITH HAPTIC SENSING AND COMPLIANT JOINTS

A Thesis
Presented to
The Academic Faculty

by

Advait Jain

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Interactive Computing

Georgia Institute of Technology
December 2012

MOBILE MANIPULATION IN UNSTRUCTURED ENVIRONMENTS WITH HAPTIC SENSING AND COMPLIANT JOINTS

Approved by:

Prof. Charles C. Kemp, Advisor
Biomedial Engineering
Georgia Institute of Technology

Prof. Magnus B. Egerstedt
Electrical and Computer Engineering
Georgia Institute of Technology

Prof. Harvey Lipkin
Mechanical Engineering
Georgia Institute of Technology

Prof. James M. Rehg
Interactive Computing
Georgia Institute of Technology

Prof. Mike Stilman
Interactive Computing
Georgia Institute of Technology

Date Approved: July 20, 2012

SUMMARY

We make two main contributions in this thesis. First, we present our approach to robot manipulation, which emphasizes the benefits of making contact with the world across all the surfaces of a manipulator with whole-arm tactile sensing and compliant actuation at the joints. In contrast, many current approaches to mobile manipulation assume most contact is a failure of the system, restrict contact to only occur at well modeled end effectors, and use stiff, precise control to avoid contact.

We develop a controller that enables robots with whole-arm tactile sensing and compliant actuation at the joints to reach to locations in high clutter while regulating contact forces. We assume that low contact forces are benign and our controller does not place any penalty on contact forces below a threshold. Our controller only requires haptic sensing, handles multiple contacts across the surface of the manipulator, and does not need an explicit model of the environment prior to contact. It uses model predictive control with a time horizon of length one, and a linear quasi-static mechanical model that it constructs at each time step.

We show that our controller enables both a real and simulated robots to reach goal locations in high clutter with low contact forces. While doing so, the robots bend, compress, slide, and pivot around objects. To enable experiments on real robots, we also developed an inexpensive, flexible, and stretchable tactile sensor and covered large surfaces of two robot arms with these sensors. With an informal experiment, we show that our controller and sensor have the potential to enable robots to manipulate in close proximity to, and in contact with humans while keeping the contact forces low.

Second, we present an approach to give robots common sense about everyday forces in the form of probabilistic data-driven object-centric models of haptic interactions. These models can be shared by different robots for improved manipulation performance. We use pulling open doors, an important task for service robots, as an example to demonstrate our approach.

Specifically, we capture and model the statistics of forces while pulling open doors and drawers.

Using a portable custom force and motion capture system, we create a database of forces as human operators pull open doors and drawers in six homes and one office. We then build data-driven models of the expected forces while opening a mechanism, given knowledge of either its class (e.g, refrigerator) or the mechanism identity (e.g, a particular cabinet in Advait's kitchen). We demonstrate that these models can enable robots to detect anomalous conditions such as a locked door, or collisions between the door and the environment faster and with lower excess force applied to the door compared to methods that do not use a database of forces.

Dedicated to my parents.

“If we could only risk a light,” he whispered.

“We have other senses than eyes,” she said.

- in Dune by Frank Herbert

The blank walls were full of surprises, all ready to reveal themselves at a touch on the panel: washstand, shitstool, mirror, desk, chair, closet, shelves.

- in The Dispossesed by Ursula K. Le Guin

ACKNOWLEDGEMENTS

This thesis has benefited tremendously from the support, advice, and help of a number of people that I am very fortunate to have as mentors, friends, and family.

First, I would like to thank my advisor, Charlie Kemp, for the deluge of ideas on how to make robots more capable, and willingness and enthusiasm to build and buy the robots, sensors, and other hardware to pursue some of them. I greatly appreciate and respect Charlie's advice on everything from writing code and papers, giving talks, to building robots.

I am also very grateful to Harvey Lipkin, James Rehg, Magnus Egerstedt, and Mike Stilman for serving on my thesis committee and for their insightful suggestions.

A warm thanks to all the fellow members and alumni of the Healthcare Robotics Lab for their friendship, help, and for making our lab an exciting place to be. A special mention to Cressel Anderson for helping me get started in lab as we worked together on the robots EL-E and Cody, Hai Nguyen for our collaboration while collecting the forces applied while opening doors, Marc Killpack for collaboration over the last year and a half on our research on reaching in clutter, and Sarvagaya Vaish for assisting with tactile sensor fabrication. Thanks also to Travis Deyle for many enjoyable conversations about robotics, science fiction, and startups.

My stay in Atlanta would have been very different without the wonderful friends that I made outside of work. A salute each, in chronological order, to Ashish Sangwan, Mayank Goel, Uday Gurnani, Rohit Moghe, Vijay Sukumaran, Kartick and Prabhakar Narasimhadevara, Ahmad Haider, Naveen Makineni, Satyan Tellikepalli, and Akanksha Prakash. I am especially grateful to Kartick and Prabhakar Narasimhadevera, Naveen Makineni, and Ahmad Haider for always being willing to have me over for a home-cooked meal, coffee, or beer.

My biggest thanks goes to my family for their love and support, which helped me to keep going. Thanks to my sister and brother-in-law for being fantastic friends. Thanks to my parents for inspiring me to try to be a good engineer and to take pride in my workmanship.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

1.1 The Big Picture

Humans and animals dramatically outperform current autonomous mobile manipulators in unstructured environments. Whole-body tactile sensing and compliant actuation are notable characteristics of biological organisms. Organisms across spatial scales, from small nematodes to insects and mammals, can sense forces across their entire bodies [21, 66, 127, 123]. Compliant actuation is also common and has inspired numerous roboticists [76, 14, 141]. Additionally, humans and animals can successfully manipulate in unstructured environments with only haptic feedback, and their performance degrades sharply in the absence haptic feedback [99, 97, 98, 36].

We draw inspiration from the actuation and sensing in these biological systems, and present approaches to mobile manipulation that use proprioception, and force sensing or tactile sensing or both, in conjunction with compliant actuation at the joints. Our goal in this thesis is to enable robots to manipulate effectively with haptic sensing and to perform well even when other sensing modalities are not well matched to the manipulation task. For example, while manipulating in cluttered environments, line of sight sensors may be of limited use. Likewise, detecting that a door is locked will be harder using non-haptic sensing. We make contributions to two areas that we describe next.

1.1.1 Manipulation with Multiple Contacts

Manipulation research has often focused on avoiding contact with the environment, except at well-modeled locations. Many current approaches attempt to model the environment with high fidelity and then perform position control with a stiff manipulator to restrict contact to occur between the end effector and a target object at predetermined locations. In essence, a majority of the manipulator's motion is intended to be free-space motion with most contact being considered a failure of the system. We contend that this paradigm, which places a high cost on contact and seeks precise kinematic control, has led to robots that dramatically underperform biological systems in unstructured environments.

Our overarching goal has been to develop a new foundation for robot manipulation that enables

robots to seek out and exploit contact with the environment. As observed in biological systems, we believe the key hardware capabilities for this will be manipulators with compliant actuation at the joints and whole-arm sensor-rich skin.

In this thesis, we take a first step towards our longer-term goal of a new foundation for mobile manipulation. We develop a controller that enables robots with whole-arm tactile sensing and compliant actuation at the joints to reach to locations in high clutter while regulating contact forces. We assume that low contact forces are benign and our controller does not place any penalty on contact forces below a threshold. Our controller only requires haptic sensing, handles multiple contacts across the surface of the manipulator, and does not need an explicit model of the environment prior to contact. We show that our controller enables both real and simulated robots to reach goal locations in high clutter with low contact forces. While doing so, the robots bend, compress, slide, and pivot around objects.

1.1.2 Data-Driven Models of Haptic Interactions

Machine intelligence has benefitted greatly from large collections of sensory data. Web-based databases of user-generated content, such as videos from YouTube, images from Flickr, and 3D models from Google 3D Warehouse, have begun to support the development of robots and enabling technology [113, 121, 115, 119, 216]. More generally, research has shown that large datasets can lead to performance gains and make computationally simple methods effective [213, 73].

Although vast quantities of captured real-world auditory and visual data are available to humans and machines, captured real-world haptic data is less common. Little is known about the statistics of real-world mechanical systems in human environments and robots currently lack common sense about the forces involved in everyday manipulation tasks.

For example, despite progress towards assistive robots capable of opening doors and drawers, the answers to basic questions have been unclear, such as, "How hard does a robot need to pull to open most doors?" and "How high does a robot need to reach to open most drawers?". Given the wide variation in the forces required to initially open a door (e.g., $\sim$60N for a spring-loaded door, $\sim$30N for a refrigerator, and $<$5N for a kitchen cabinet), a robot without common sense about everyday forces risks damaging a locked door or giving up prematurely on a functioning door. Likewise, while opening a door for someone, distinguishing forces due to the door opening properly versus the door colliding with the person could improve safety.

In this thesis, we present data-driven methods to inform robots about the forces that they are likely to encounter when performing specific tasks. In the context of door opening, we demonstrate

that data-driven models can be used to haptically recognize specific doors, haptically recognize classes of door (e.g., refrigerator vs. kitchen cabinet), and better detect haptic events (e.g., a locked door or collision), even when opening a specific door for the first time. We also show that two distinct robots can use forces captured from people opening doors to better detect haptic events. This illustrates the potential for data-driven models of task-specific forces captured by people and robots to inform robot manipulation.

In general, this type of common sense knowledge would help robots behave more intelligently. In the future, robots might use these data in numerous ways, including selecting better postures prior to manipulation, detecting when mechanisms are in need of repair, and anticipating when a human will require assistance. Data-driven models of haptic interactions may enable robots to efficiently detect anomalous conditions without excessive force. For example, a robot may be able to haptically detect that it is attempting to insert the incorrect key or that the door is not completely shut. Likewise, a robot may stop inserting a USB plug, flip it, and retry.

This approach could also be relevant to tasks related to activities of daily living such as bed baths, shaving, grooming, and manipulating a person's body. Moreover, these data could be used by humans to rationally design robots with the kinematic and force capabilities necessary to perform real-world tasks.

1.2 Robots

We use two mobile manipulators in this research, Cody and a Willow Garage PR2, shown in fig. 1 on the following page. During the course of this research, we have made changes to the robots including additional sensors and different end effectors. We describe these changes in the different chapters of this thesis.

1.2.1 Cody

The members of the Healthcare Robotics Lab built Cody using off-the-shelf components and some custom parts. It consists of two 7 degrees of freedom (DoF) arms from MEKA Robotics (MEKA A1), an omnidirectional mobile base from Segway (RMP 50 Omni), and a 1-DoF linear actuator from Festo that can lift the manipulator and sensors from ground level to 1.2m above the ground.

The arms use series elastic actuators (SEAs) [167] at each degree of freedom and have an ATI Mini40 force-torque sensor at the wrist. For our initial work with Cody, we did not have reliable odometry due to the large amounts of slip in the mecanum wheels. In later work we control the mobile base with a PID controller that uses visual odometry as we describe in [111].

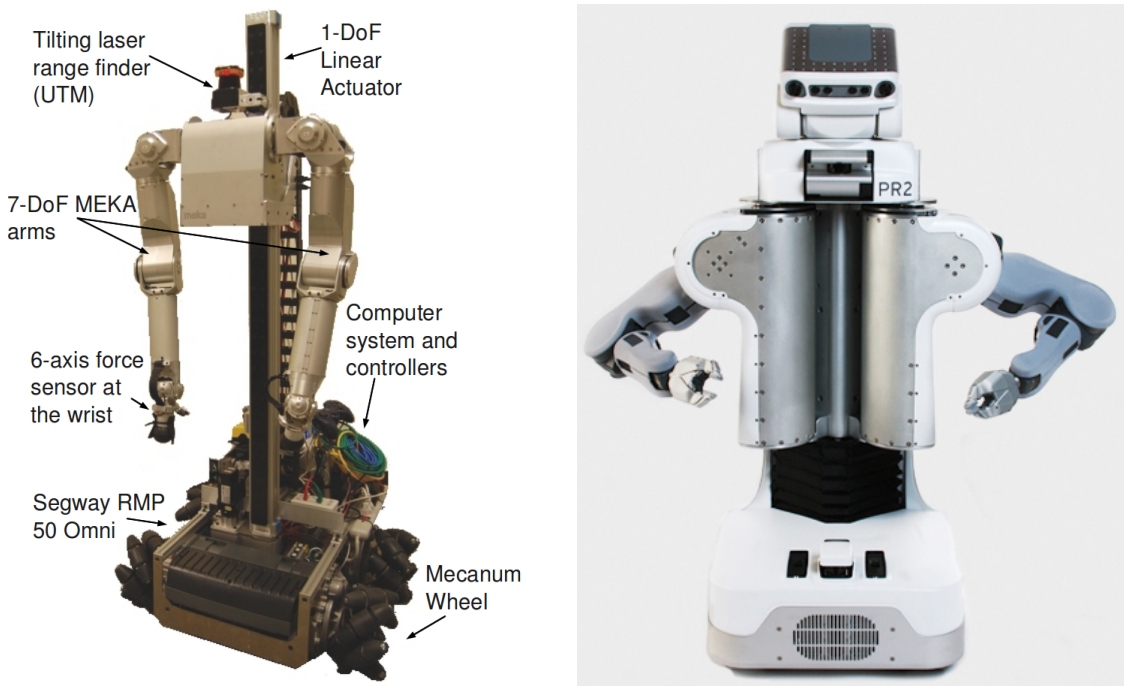


Figure 1: ***Left:*** *The mobile manipulator Cody (in 2009), that members of the Healthcare Robotics Lab built using off-the-shelf and some custom components.* ***Right:*** *A Willow Garage PR2. During the course of this research, we have made changes to the two robots including additional sensors and different end effectors. We describe these changes in the different chapters of this thesis.*

1.2.2 Willow Garage PR2

The Willow Garage Personal Robot 2 (PR2) has two 7-DoF arms with springs that provide passive gravity compensation by counterbalancing the weight of the upper and lower arm links. As a result the motors in the joints do not have to apply torques to overcome gravity and have low reduction gears that permit compliant actuation via current control [225]. The PR2 also has an omnidirectional mobile base and a telescoping spine that can lift the arm and sensors by 31cm.

1.3 Organization of this Thesis

The rest of this thesis is organized as follows.

- In chapter 2 on page 7, we present our approach to robot manipulation, which emphasizes the benefits of making contact with the world across all the surfaces of a manipulator. We develop a controller that enables robots with whole-arm tactile sensing and compliant actuation at the joints to reach to locations in high clutter while regulating contact forces. We assume that low contact forces are benign and our controller does not place any penalty on contact forces below a threshold. Our controller uses model predictive control with a time horizon of length

one, and a linear quasi-static mechanical model that it constructs at each time step.

- In chapter 3 on page 24, we evaluate our model predictive controller in software simulation and on a real robot and show that the controller enables both real and simulated robots to reach goal locations in high clutter with low contact forces. While doing so, these robots push aside movable objects, deform compliant objects, and perceive the world through contact. Our tests include a real robot reaching into a cinder block and simulated foliage using a novel tactile sensor array on its forearm. In simulation, we also show that our controller performs better with whole-arm tactile sensing than with a force-torque sensor at each link, and that the relative value of whole-arm tactile sensing increases with higher clutter.

- In chapter 4 on page 46, we present our design for tactile sensor arrays using conductive and resistive fabric. Our design is inspired by rSkin, an open source project, and is relatively straightforward to replicate. We cover the end effector, wrist joints, and forearm of Cody, and large surfaces of a PR2 arm with tactile sensor arrays with low spatial resolution. With an informal experiment, we show that our controller of chapter 2 and this sensor have the potential to enable robots to manipulate in close proximity to, and in contact with humans. We also present a single trial of manipulating an object on a shelf with multiple contacts between the robot arm and the environment.

- In chapter 5 on page 63, we present a set of feedback controllers that enable a mobile manipulator to reliably and autonomously approach and open doors and drawers for which only the location and orientation of the handle have been provided. The robot uses low stiffness compliant actuation at the joints and a wrist-mounted force-torque sensor.

  We empirically demonstrate that our controllers are robust with respect to variations in the mechanism, the pose of the base, the stiffness of the arm, the friction between the door handle and the end effector, and the way the handle was hooked. We also demonstrate that our controllers can coordinate the movement of the robot's omnidirectional base and compliant arm while pulling open a door or drawer.

- In chapter 6 on page 99, we present a custom force and motion capture system that we built to capture the forces and trajectories while pulling open doors and drawers. We demonstrate that the forces while opening these seemingly simple everyday devices have complex structure that is hard to explain with mechanical models but can be captured and characterized using data-driven methods.

- In chapter 7 on page 116, we present data-driven methods to inform robots about the forces that they are likely to encounter when performing specific tasks. In the context of door opening, we demonstrate that data-driven models can be used to haptically recognize specific doors, haptically recognize classes of door (e.g., refrigerator vs. kitchen cabinet), and better detect haptic events (e.g., a locked door or collision), even when opening a specific door for the first time. We also show that two distinct robots can use forces captured from people opening doors to better detect haptic events. This illustrates the potential for databases of task-specific forces to be captured by people and robots to inform robot manipulation.

- In appendix A on page 147, we provide links to code, data, videos, and hardware designs for different chapters of this thesis.

CHAPTER II

REACHING IN CLUTTER WITH WHOLE-ARM TACTILE SENSING: CONTROLLER DEVELOPMENT

In this chapter, we first present our approach to robot manipulation, which emphasizes the benefits of making contact with the world across the entire manipulator. We assume that low contact forces are benign, and focus on the development of robots that can control their contact forces during goal-directed motion. We also assume that the robot to be controlled has low-stiffness actuation at its joints, and tactile sensing across the entire surface of its manipulator. We then present a novel controller that exploits these assumptions. The controller only requires haptic sensing, handles multiple contacts across the surface of the manipulator, and does not need an explicit model of the environment prior to contact. It uses model predictive control (MPC) with a time horizon of length one, and a linear quasi-static mechanical model that it constructs at each time step.

The first working version of the controller described in this chapter was developed by Prof. Charles C. Kemp. We describe a modified version in this chapter which also appears in an article that is currently under review [95].

The rest of this chapter is organized as follows. Section 2.1 on the following page describes our approach for manipulation in clutter. We then discuss related research in section 2.2 on page 11 and derive our model predictive controller in section 2.3 on page 14.

In the next chapter (Chapter 3 on page 24), we show that this controller enables both a real and simulated robot to reach goal locations in high clutter with low contact forces. While doing so, these robots push aside movable objects, deform compliant objects, and perceive the world through contact.

2.1 Overview of Our Approach

Research on robot manipulation has often emphasized collision free motion with occasional contact restricted to the robot's end effector. In essence, most of the manipulator's motion is intended to be free-space motion and unintended contact is considered to be a failure of the system. In contrast, animals often appear to treat contact between their arms and the world as a benign event that does not need to be avoided. For example, humans make extensive contact with their forearms even during mundane tasks, such as eating or working at a desk.

In this and the following chapter, we present progress towards new foundational capabilities for robot manipulation that take advantage of contact across the entire arm. Our primary assumption is that, for a given robot, environment, and task, contact forces below some value have no associated penalty. For example, when reaching into a bush, moderate contact forces are unlikely to alter the robot's arm or the bush in undesirable ways. Likewise, even environments with fragile objects, such as glassware on a shelf, can permit low contact forces. While some situations merit strict avoidance of contact with an object, we consider these to be rare, and instead focus on control methods that allow contact.

Our main contribution is a novel controller that enables a robot arm to move within an environment while regulating contact forces across its entire surface. The controller assumes that contact forces can be sensed across the surface of the arm, and that the arm's joints can be modeled as linear torsional springs. The controller uses model predictive control (MPC) with a time horizon of length one and a linear quasi-static mechanical model. At each time step, the controller constructs a model and solves an associated quadratic programming problem that minimizes the predicted distance to a goal subject to constraints on the predicted changes in contact forces.

Figure 2: ***Left:*** *View of foliage from the robot's perspective. Two rigid blocks of wood are occluded by the leaves.* ***Right:*** *Image of the robot after it has successfully reached the goal location using the controller we present in this chapter. The red circle denotes the position of the end effector.*

(a) (b) (c) (d) (e)

Figure 3: *Animals reach into clutter while foraging. (a) A raccoon reaches into a bird house to find eggs and young [5]. (b) A Long-tailed Macaque grasps fruit in dense foliage [3]. (c) When noodling, people find catfish holes from which to pull fish out [4]. (d)-(e) A person makes contact along his forearm while reaching for objects in a cluttered cabinet and refrigerator. (All images used with permission)*

We empirically evaluated our controller's performance with respect to the task of haptically reaching to a goal location in high clutter (see fig. 2 on the preceding page). The clutter could consist of a variety of fixed, movable, and deformable objects, and the robot was not given a model of the environment.

As described in chapter 3 on page 24 we used a simulated robot, a real robot with simulated tactile sensing, and a real robot with real tactile sensing across its forearm for our tests. All of the robots successfully reached to locations in clutter while regulating contact forces across their arms. While doing so, the robots performed maneuvers, such as pushing into a compliant object until the contact force was too high, and then pivoting around it. In an automated test with simulated foliage, our controller outperformed a baseline controller in terms of success rate and low contact forces. In simulation, our controller had higher performance with whole-arm tactile sensing than with per-link force-torque sensing.

2.1.1 Biological Inspiration

Animals have inspired our approach with their adept manipulation in cluttered environments. As illustrated in fig. 3, animals reach for objects in high clutter with poor visibility when foraging for food [49, 86, 32]. While doing so, they are able to handle contact at multiple locations across their arms, and often appear to treat contact as a benign event.

Whole-body tactile sensing and compliant actuation are notable characteristics of biological organisms. Organisms across spatial scales, from small nematodes to insects and mammals, can sense forces across their entire bodies [21, 66, 127, 123]. Compliant actuation is also common and

has inspired numerous roboticists [76, 14, 141].

2.1.2 Benefits of Contact, Whole-body Tactile Sensing, and Low-stiffness Actuation

Whole-body tactile sensing and low-stiffness actuation have potential benefits for robots manipulating in clutter, as does allowing contact between the robot and the environment.

Allowing contact with the arm can increase the effective range of motion of the manipulator. As illustrated in fig. 4, the performance loss due to avoiding contact is exacerbated by safety margins and an inability to apply forces that compress or move objects. Similarly, if the robot has a compliant exterior, avoiding contact forfeits the additional range of motion achievable by compressing this exterior.

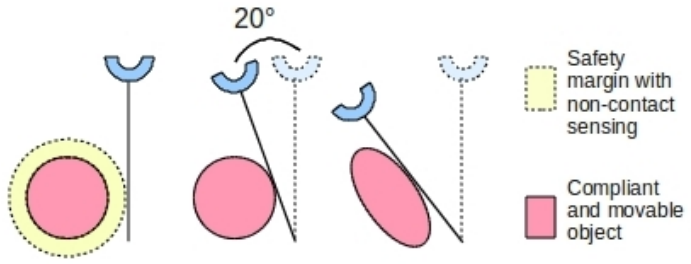


Figure 4: *Example illustrating the available range of motion for a 1 degree-of-freedom (DoF) arm if the controller uses a safety margin with non-contact sensing* ***(left)****, allows contact with an object* ***(middle)****, and allows the arm to push into compliant and movable objects* ***(right)****.*

Whole-body tactile sensing has advantages in terms of distinguishing between distinct force distributions and contact configurations. For example, a force distributed over a small area results in higher stress with greater potential for damage than the same force distributed over a large area. Also, the geometries associated with distinct contact regions, such as a rigid point, line, or plane, imply distinct options for movement. Prior research has attempted to use the geometry of links, joint torques, and measurements from force-torque sensors to estimate contact properties (e.g, [53, 24, 104, 41]). However, interpretation of data from these sensors can often be ambiguous in multi-contact situations [183]. In practice, the estimation can also be sensitive to the state of the manipulator, the fidelity of the torque estimates, and friction and flexibility at the joints [47, 54].

Robotics researchers have demonstrated that low-stiffness joints lower interaction forces during incidental contact and can be beneficial for unmodeled interactions [88, 167, 30, 166]. Within rigid clutter, low-stiffness at the joints may also mitigate problems due to jamming and wedging, much like compliance can be beneficial in peg-in-hole problems [52, 219, 218, 134].

2.1.3 Challenges Associated with Reaching in High Clutter

We focus on reaching to a goal location in high clutter, which entails challenges such as:

- *Lack of non-contact trajectories:* As clutter increases, approaches that avoid contact with the environment will have a diminishing set of feasible trajectories.
- *Contact with only the end effector may be inefficient or infeasible:* Removing or rearranging the clutter by serially making contact with the end effector may be inefficient or infeasible.
- *Creating detailed geometric models in advance may be infeasible:* Clutter can consist of unique objects and configurations that have not been encountered before.
- *Observation of geometry is obstructed:* Occlusion can prevent conventional line of sight sensors, such as cameras and laser range finders, from observing the geometry of the clutter in advance (see fig. 2 on page 8).
- *Mechanics are difficult to infer without contact:* Non-contact sensing provides limited ability to infer the mechanical properties of the clutter, such as whether or not an object can be bent or moved out of the way.

Notably, many approaches to manipulation are poorly matched to the challenges of high clutter. For example, approaches that rely on preexisting detailed models, estimation of models via conventional line-of-sight sensing, or collision-free motions with the arm would fare poorly in many high-clutter real-world situations (e.g., [107, 203, 200, 188]).

In contrast, our approach directly addresses these challenges by allowing multiple contacts across the entire surface of the arm, not requiring a detailed model of the environment prior to contact, and only requiring contact-based sensing.

2.2 Related Work

2.2.1 Manipulation in Clutter

In this paper, we focus on the task of reaching to a goal location in high-clutter. Previous robotics research has addressed the tasks of generating collision free trajectories (e.g, [126, 122, 107]), generating reaching motions in free space (e.g, [139, 204, 75]), and manipulating objects in uncluttered environments (e.g, [146, 93, 81, 153, 179, 188]).

Most prior research on manipulation in clutter restricts contact between the robot and its environment to the end effector. [124] presents a method to assist teleoperators that finds collision

free poses for the end effector in clutter. [46, 203, 202] present motion planners that use models of objects in the world to enable robots to rearrange clutter by pushing, grasping, and moving objects with their end effectors. The planned trajectories are executed without sensor feedback. [135] describes a simple end effector design that can grasp a single marker from a cluttered pile of markers, and haptically estimate its pose in the gripper. Although the task is different, aspects of the approach are similar in spirit to ours: the robot interacts with high clutter, does not use a detailed model of the environment prior to contact, allows multiple contacts across the surfaces of the end effector, uses haptic sensing, and retries several times.

2.2.2 Multi-contact Manipulation

[151] presents a framework for controlling a robot with multiple contacts along the links. It generalizes previous direct force control methods [172, 110] to allow for contacts at points other than the end effector. It also considers the dynamic stability of the robot. This method requires a full dynamic model of the robot, models contacts as being rigid, and assumes that the robot has at least six degrees of freedom (DoF) for each contact to control the contact force and torque vector [193]. Using this framework, results have been shown in simulation [192, 193], and on a real robot in relatively controlled settings [159]. No results have been shown in cases where the robot makes additional unpredicted contact with the environment or loses contact at some locations.

Researchers have demonstrated that a humanoid robot can make contact at multiple, predetermined locations on its body to better achieve dynamic stability while performing tasks, such as taking a large step [125], climbing a ladder [74], and sitting down in a chair [59]. These approaches require a complete geometric model of the world, assume rigid contacts, and do not incorporate sensor feedback as the robot executes the planned kinematic trajectory.

Other work on multi-contact manipulation exists in the context of using the surfaces of a multi-fingered hand, or the entire body to grasp and manipulate objects (e.g, [22, 23, 83, 162, 82]). Researchers in this area often focus on the task of finding and achieving stable grasps for objects.

Research on whole-arm collision detection has used a model of the dynamics of the robot arm to detect deviations from an expected torque to detect collisions across the entire arm of a robot (e.g, [40, 39, 71]). Most of this work has been limited to contact at a single location and maneuvers to move away from the estimated contact location.

2.2.3 Tactile Exploration

Recent work on tactile exploration has focused on exploring unmodeled environments or unknown objects in the environment. [137] designed a manipulator for tactile exploration of oil wells, and [168] developed a biomemetic vibrissal sensor for robot navigation and object detection. Other researchers have focused on tactile exploration of objects with the robot's end effector for object recognition [25, 161], object localization [157], and estimation of object properties [33].

2.2.4 Motion Planning with Deformable Objects

Manipulation research often assumes that objects that the robot interacts with are rigid. There is some research on motion planners that allow the robot to make contact with, and push into deformable objects (e.g. [61, 177, 154]). These approaches assume knowledge of the specific configuration of the objects and require accurate and detailed models of how the objects deform.

2.2.5 Robot Locomotion

Our approach to robot manipulation has similarities to approaches that have been successful for robot locomotion. For example, researchers have developed robots that locomote in cluttered environments without detailed geometric models of the terrain nor planning over long time horizons [186, 171]. Likewise, whole body contact, and contact in general, has not been considered undesirable. For example, robots have used contact all over their bodies to traverse the ground and swim in granular media [138, 130]. Additionally, the use of simple mechanical models, compliance, and force sensing is common for robot locomotion [171, 166, 88, 64].

2.2.6 Model Predictive Control

One of the initial application areas for model predictive control (MPC) was chemical process control [63]. It is often referred to as receding horizon control when used for control of aerial vehicles [20, 12]. MPC has also been used in robot locomotion research (e.g, [220, 131, 58]). In terms of robot manipulation, MPC has recently been used for bouncing a ball [120], generating manipulator trajectories to compensate for inertial forces on a boat [62], controlling a 6 DoF cable-driven parallel manipulator [50], and reaching in free space [85].

2.3 Model Predictive Controller

The controller that we have developed uses linear model predictive control (MPC) with a time horizon of length one. Specifically, using the notation of [142], our controller uses a linear discrete-time model of the system,

$$x(k+1) = Ax(k) + Bu(k), \tag{1}$$

where $x(k)$ is the state and $u(k)$ is the control input.

At each time step, k, the controller computes a sequence of control inputs, $u^*(i), i = k \ldots (k + N - 1)$, to minimize a quadratic objective function of $x(k), \ldots, x(k+N)$ and $u(k), \ldots, u(k+N-1)$, subject to linear inequality constraints on $x(k), \ldots, x(k+N)$ and $u(k), \ldots, u(k+N-1)$, where N is the length of the time horizon of the model predictive controller. This defines a quadratic program [142]. The controller then uses only the first control input by setting $u(k) = u^*(k)$, and reformulates the quadratic program at the next time step. In this paper, we use a time horizon of length one ($N = 1$), and recompute the A and B matrices in eq. (1) on the current page at each time step.

In the rest of this section, we describe our model predictive controller for manipulation with multiple contacts. First, in section 2.3.1 on this page, we give an overview of the controller that we have developed. Next, we present the hierarchy of controllers running on our robots in section 2.3.2 on the next page. In section 2.3.3 on page 16, we describe the linear quasi-static model that our model predictive controller uses, and detail the quadratic program that the controller solves at each time step section 2.3.4 on page 19. We then describe extensions to the quadratic program in section 2.3.5 on page 21.

2.3.1 Overview of the One-Step Model Predictive Controller

The model predictive controller attempts to move the end effector along a straight line to the goal, subject to constraints on the predicted contact forces. It explicitly allows the robot to apply any force less than a *don't care force threshold*, $f_{c_i}^{thresh}$, at each contact, c_i.

Our controller has the following parameters that influence its behavior:

- *Goal location* ($x_g \in \Re^3$)*:* This is the location that the controller attempts to move the end effector to.

- *Contact stiffness matrices* ($K_{c_i} \in \Re^{3\times3}$)*:* These are the controller's estimates of the stiffness matrices for each contact location along the arm. In this paper, we assume that the stiffness

at each contact is non-zero along the direction of the contact force sensed by the real or simulated sensors and is zero in the other directions.

- *Don't care force thresholds* ($f_{c_i}^{thresh} \in \Re^3$)*:* The controller attempts to keep the force at each contact below this value, and there is no penalty or constraint for contact forces below this threshold.

- *Maximum rate of change of contact force* ($\Delta f_{c_i}^{rate} \in \Re^3$)*:* This term limits the predicted change in the contact force over one time step.

- *Safety force threshold* ($f_{c_i}^{safety} \in \Re^3$)*:* If the contact force, f_{c_i}, exceeds this safety threshold value, the MPC controller stops and we report it as a failure.

2.3.2 Control Structure

Figure 5 shows our hierarchical control structure. A joint space impedance controller runs at 1kHz. Our implementation of the model predictive controller runs in Python on a conventional desktop computer at a rate that varies between approximately 50Hz and 100Hz. Some of this variation depends on the number of contact locations.

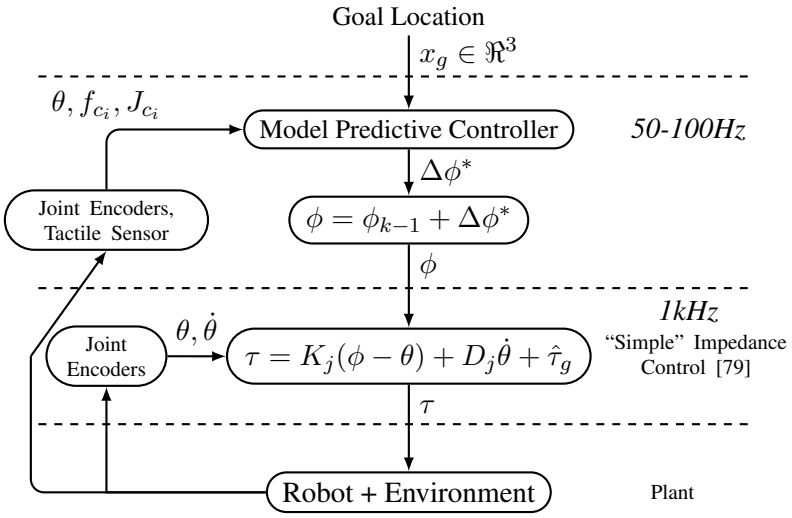


Figure 5: *Block diagram showing the hierarchical control structure and the equations of motion. The controller frequencies are specific to our implementation.*

2.3.2.1 "Simple" Impedance Controller

For a detailed description and analysis of this form of joint space impedance control, we refer the reader to [79]. The input to the 1 kHz "simple" impedance controller, ϕ, is called a virtual trajectory.

The controller uses feedback from the joint encoders to command torques at the joints, τ, that are given by

$$\tau = K_j(\phi - \theta) + D_j\dot{\theta} + \hat{\tau}_g(\theta). \tag{2}$$

For this paper, K_j and D_j are constant $m \times m$ diagonal joint-space stiffness and damping matrices, $\theta \in \Re^m$ and $\dot{\theta} \in \Re^m$ are the current joint angles and joint velocities, and $\hat{\tau}_g \in \Re^m$ is a gravity compensating torque vector that is a function of θ. The robot arm has m joints.

As a result, the closed loop system behaves as if the arm is connected to the joint-space virtual trajectory, ϕ, via torsional visco-elastic springs at the joints. If ϕ is held constant, "simple" impedance control can be shown to result in stable interaction with passive environments for contacts all over the arm [79, 78].

"Simple" impedance control, sometimes referred to as equilbrium point control, does not explicitly model the dynamics of the arm, unlike other approaches to force control and impedance control [193, 13]. A model of the dynamics could potentially improve the performance of our robots. In practice, we have found "simple" impedance control to be stable, reliable, and sufficient to reach into clutter and perform other tasks [56, 57, 91, 92, 94]. It also has the advantage of being straightforward to implement without the need for system identification. Other researchers have looked at similar robotic control strategies in simulation [70], in free-space motions [222], in legged locomotion [140], and in rhythmic manipulation from a fixed base [223].

2.3.2.2 Model Predictive Controller

The input to the model predictive controller is a goal location, $x_g \in \Re^3$. The controller uses feedback from the joint encoders and whole-arm tactile sensing to compute $\Delta\phi^* \in \Re^m$, an incremental change in the virtual joint-space trajectory. This $\Delta\phi^*$ is the control input, $u(k)$, of eq. (1) on page 14.

2.3.3 Linear Discrete-Time Model

In this section, we derive a linear discrete-time quasi-static model, similar to eq. (1), for the arm and its interaction with the world. The model is of the form

$$\theta(k+1) = \theta(k) + B\Delta\phi(k), \tag{3}$$

where $\theta \in \Re^m$ is the state of the system (vector of joint angles for a robot with m joints), the control input $\Delta\phi \in \Re^m$ is the incremental change in the joint-space virtual trajectory of the impedance controller, and $B \in \Re^{m \times m}$.

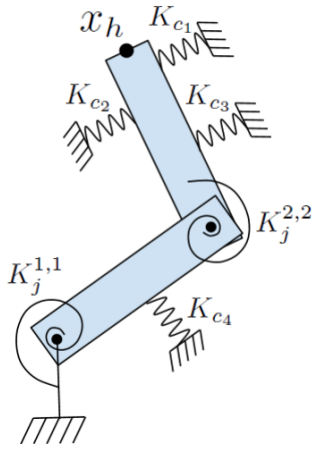


Figure 6: *Graphical representation of a planar version of the quasi-static mechanical model with torsional springs at the joints of the robot and linear springs at contacts that our model predictive controller uses, as we describe in section 2.3.3 on the previous page.*

We begin by assuming that the robot has a fixed and statically stable mobile base and the arm is in contact with the world at n locations. We denote the i^{th} contact as c_i. The equations of motion in joint space are

$$M(\theta)\ddot{\theta} + C(\theta, \dot{\theta})\dot{\theta} + \sum_{i=1}^{n} J_{c_i}^T(\theta) f_{c_i} + \tau_g(\theta) = \tau, \tag{4}$$

where $M \in \Re^{m \times m}$ is the inertia matrix, $C \in \Re^{m \times m}$ is the Coriolis matrix, $J_{c_i} \in \Re^{3 \times m}$ is the Jacobian matrix for the i^{th} contact, $f_{c_i} \in \Re^3$ is the force at contact c_i, $\tau_g \in \Re^m$ is the vector of torques due to gravity at each joint, and $\tau \in \Re^m$ is the vector of torques applied by the actuators at the joints. Equation (4) ignores effects such as friction at the joints, but is commonly used in robotics [60].

Combining the equations of motion (eq. (4)) with the impedance control law (eq. (2) on the preceding page) gives the model of the arm and its interaction with the world,

$$M\ddot{\theta} + C\dot{\theta} + \sum_{i=1}^{n} J_{c_i}^T f_{c_i} + \tau_g = K_j(\phi - \theta) + D_j\dot{\theta} + \hat{\tau}_g. \tag{5}$$

In this thesis, as an approximation, we assume that the dynamics are negligible since we are moving slowly, and that the gravity compensating torques are perfect. We remove all terms with $\ddot{\theta}$ or $\dot{\theta}$ from eq. (5), and set $\hat{\tau}_g = \tau_g$ to get

$$\sum_{i=1}^{n} J_{c_i}^T f_{c_i} = K_j(\phi - \theta), \tag{6}$$

which is a quasi-static model. These approximations help us obtain a linear discrete time model that results in a computationally favorable quadratic program. In eq. (6) on the previous page, the torques at the joints due to the contact forces (left-hand side) balance the torques applied by the actuators in the joints (right-hand side).

For the contact model, we ignore friction at the contacts and assume that each contact behaves like a linear spring normal to the surface of the robot arm. These assumptions are similar to the Hertzian contact model [105, 100]. This results in a mechanical model with torsional springs at the joints and linear springs at the contacts as illustrated in fig. 6 on the preceding page.

If we take the difference between eq. (6) on the previous page at times k and $k+1$, we obtain

$$\sum_{i=1}^{n} J_{c_i}^T(k+1) f_{c_i}(k+1) - J_{c_i}^T(k) f_{c_i}(k) = K_j(\phi(k+1) - \phi(k) - \theta(k+1) + \theta(k)). \quad (7)$$

We assume that the change in the configuration of the arm in one time step, $\theta(k+1)-\theta(k)$, is small and we approximate $J_{c_i}(k+1)$ with $J_{c_i}(k)$. This reduces eq. (7) to

$$\sum_{i=1}^{n} J_{c_i}^T(k)(f_{c_i}(k+1) - f_{c_i}(k)) = K_j(\Delta\phi(k) - \theta(k+1) + \theta(k)), \quad (8)$$

where $\Delta\phi(k) = \phi(k+1) - \phi(k)$ is the control input of the model predictive controller, see eq. (3) on page 16 and fig. 5 on page 15.

Using the linear elastic spring model for the contacts,

$$f_{c_i}(k+1) - f_{c_i}(k) = K_{c_i} J_{c_i} \Delta\theta(k), \quad (9)$$

where $\Delta\theta(k) = \theta(k+1) - \theta(k)$. We can now use eq. (9) to rewrite eq. (8) as

$$\left(\sum_{i=1}^{n} J_{c_i}^T K_{c_i} J_{c_i}\right) \Delta\theta(k) = K_j \left(\Delta\phi(k) - \Delta\theta(k)\right). \quad (10)$$

Equation (10) can be interpreted as the equations for quasi-static motion. The left hand side is the change in the torques applied by the torsional visco-elastic springs at the joints due to an incremental update to the virtual trajectory, after the configuration of the arm changes by an amount $\Delta\theta(k)$. The right hand side is the change in the joint torques due to the linear springs at the contacts. Quasi-static motion implies that the system is in static equilibrium and these two torque vectors are equal.

We now rearrange eq. (10) to get

$$\theta(k+1) = \theta(k) + \left(K_j + \sum_{i=1}^{n} J_{c_i}^T K_{c_i} J_{c_i}\right)^{-1} K_j \Delta\phi(k). \quad (11)$$

$\left(K_j + \sum_{i=1}^{n} J_{c_i}^T K_{c_i} J_{c_i}\right)$ is the sum of a positive definite matrix, K_j, and positive semi-definite matrices, $J_{c_i}^T K_{c_i} J_{c_i}$, and is therefore positive definite and invertible.

Equation (11) on the preceding page is the linear discrete-time model of the system that our controller generates and uses at each time step. The controller computes the parameters for this model, J_{c_i} and K_{c_i}, based on the current contact locations and a stiffness for each contact. Note that the contact forces, f_{c_i}, do not appear in this model, which relates change in the virtual trajectory to change in the joint angles. The contact forces do appear in the inequality constraints of the quadratic program, and are used in a quadratic penalty term if they exceed the *don't care force threshold*, $f_{c_i}^{thresh}$. In practice, measured forces are also used to detect contact and estimate contact locations. Detecting contact is especially important, since false positives would result in the controller hallucinating contact and modeling it with a spring.

Equation (11) is in the same form as eqs. (1) and (3) on pages 14–16. As we describe next, its linear form allows us to frame the optimization as a quadratic program (QP) that can be solved efficiently at each time step.

2.3.4 Quadratic Program to Compute $\Delta\phi^*$

Using the terminology of [27], our optimization variable is $\Delta\phi$, an incremental change in the joint-space virtual trajectory, and we minimize a quadratic objective function subject to linear equality and inequality constraints. We use the open source OpenOpt framework to solve this quadratic program [118].

Our objective function is of the form

$$\sum_i \alpha_i g_i, \tag{12}$$

where g_i are quadratic functions of the optimization variable $\Delta\phi$, and α_i are empirically tuned scalar weights. We set $\alpha_1 = 1$ and list the values of the other scalar weights in section 2.3.5 on page 21.

We set up the quadratic program such that the solution, $\Delta\phi^*$, will result in the predicted position of the end effector that is closest to a desired position subject to constraints on the predicted change in the joint angles and contact forces. Below are the components of this quadratic program.

2.3.4.1 Move to a Desired Position

The first term of the quadratic objective function of eq. (12) attempts to move the end effector to a desired position. It is of the form

$$g_1 = \|\Delta x_d - \Delta x_h\|^2, \tag{13}$$

where $\Delta x_h = x_h(k+1) - x_h(k)$ is the predicted motion of the end effector (hand) and $\Delta x_d \in \Re^3$ is the desired change in the end effector position in one time step. The controller attempts to move the end effector in a straight line towards the goal, $x_g \in \Re^3$, with

$$\Delta x_d = \begin{cases} d_w \frac{x_g - x_h}{\|x_g - x_h\|} & if \ \|x_g - x_h\| > d_w \\ x_g - x_h & if \ \|x_g - x_h\| \leq d_w \end{cases}, \tag{14}$$

where d_w is a small constant distance. We assume the change in the joint angles will be small, and use the kinematic relationship

$$\Delta x_h = J_h \Delta\theta, \tag{15}$$

where $J_h \in \Re^{3\times m}$ is the Jacobian at the end effector (or hand), and $\Delta\theta = \theta(k+1) - \theta(k)$ is the change in the joint angles predicted by eq. (11) on page 18. We express the objective function g_1 as a quadratic function of $\Delta\phi$:

$$g_1 = \left\| \Delta x_d - J_h \left(K_j + \sum_{i=1}^{n} J_{c_i}^T K_{c_i} J_{c_i} \right)^{-1} K_j \Delta\phi \right\|^2. \tag{16}$$

2.3.4.2 *Joint Limits*

We also add two linear inequality constraints to keep the predicted joint angles within the physical joint limits. These are of the form

$$\Delta\theta_{min} \leq \Delta\theta \leq \Delta\theta_{max}, \tag{17}$$

where $\Delta\theta_{min}$ and $\Delta\theta_{max}$ are the difference between the minimum and maximum joint limits and the current configuration of the robot. Using eq. (11) on page 18 we can rewrite the inequalities of eq. (17) as linear inequalities in $\Delta\phi$.

2.3.4.3 *Contact Forces*

For each contact, we attempt to restrict the contact force f_{c_i} to be below a *don't care force threshold*, $f_{c_i}^{thresh}$, and limit the predicted change of the contact force, $\Delta f_{c_i} = f_{c_i}(k+1) - f_{c_i}(k)$, in one time step. This results in two inequalities for each contact,

$$\Delta f_{min} \leq \Delta f_{c_i} \leq \Delta f_{max}, \text{ where} \tag{18}$$

$$\Delta f_{min} = -f_{c_i}^{rate}, \text{ and} \tag{19}$$

$$\Delta f_{max} = min\left(f_{c_i}^{rate}, f_{c_i}^{thresh} - f_{c_i} \right). \tag{20}$$

$f_{c_i}^{rate}$ is a threshold on the maximum allowed predicted change in the contact force in one time step. We limit the maximum change to keep the motion of the arm smooth. For simplicity, we used the same value of $f_{c_i}^{rate}$ for both increasing and decreasing the contact force, but this is not a requirement. The term $(f_{c_i}^{thresh} - f_{c_i})$ in eq. (20) on the preceding page explicitly allows contact forces below $f_{c_i}^{thresh}$.

From eqs. (9) and (11) on page 18, the inequalities of eq. (18) on the preceding page can be expressed as linear inequalities in $\Delta\phi$.

2.3.5 Extensions to the Quadratic Program

In this section, we describe three extensions to the quadratic program of the previous section (section 2.3.4 on page 19).

2.3.5.1 Squared Magnitude of $\boldsymbol{\Delta\tau}$

To discourage large changes in the joint torques in one time step, we add a term

$$\begin{aligned} g_2 &= \|\Delta\tau\|^2 \\ &= \Delta\phi^T K_j^T K_j \Delta\phi, \end{aligned} \tag{21}$$

to the objective function after multiplying it with a scalar weight α_2, see eq. (12) on page 19. For the work we report in this paper, we set $\alpha_2 = 0.00001$ for both the real and simulated robot. Increasing this weight is useful in preventing large motions of redundant degrees of freedom. Making α_2 too large discourages all motion of the arm.

2.3.5.2 Decrease Contact Forces Above Don't Care Threshold

Due to modeling errors, the contact forces can go above the *don't care force threshold* ($f_{c_i} > f_{c_i}^{thresh}$). This can lead to an infeasible quadratic program. For example, Δf_{max} can become less than Δf_{min}.

For each contact with $f_{c_i} > f_{c_i}^{thresh}$, we replace eq. (20) on the previous page with $\Delta f_{max} = 0$. This requires the predicted contact force to either decrease or remain the same. Leaving the predicted contact forces the same by not changing the virtual trajectory, $\Delta\phi = 0$, is always feasible.

We also add an additional term g_3 to the objective function that encourages the controller to decrease these contact forces.

$$g_3 = \sum_i \left\| \Delta f_{c_i}^d - \Delta f_{c_i} \right\|^2 \text{ if } f_{c_i} > f_{c_i}^{thresh}, \tag{22}$$

where $\Delta f_{c_i}^d$ is the desired change for the contact force in one time step and Δf_{c_i} is the change in the predicted contact force. We set $\Delta f_{c_i}^d$ as a force with a constant magnitude and a direction opposite to f_{c_i}. Using eqs. (9) and (11) on page 18, we can express g_3 as a quadratic function of $\Delta\phi$.

We set the scalar weight $\alpha_3 = 1$ and $\left\|\Delta f_{c_i}^d\right\| = 0.2N$. We found that these two parameter values enabled the controller to reduce the force in a controlled manner in software simulation and on the real robot.

2.3.5.3 Limits on the Virtual Trajectory

On the robot Cody, described in section 3.1.2 on page 26, the joint-space impedance controller limits the virtual trajectory to be within the physical joint limits. To account for this, we add two additional linear constraints on $\Delta\phi$:

$$\Delta\phi_{min} \leq \Delta\phi \leq \Delta\phi_{max}. \tag{23}$$

2.3.6 Stopping Our Model Predictive Controller

In the event of a contact force above the safety threshold, $f_{c_i}^{thresh}$, our model predictive controller stops sending changes to the virtual trajectory, while the "simple" impedance controller continues to run. Theoretically, if the environment is passive, the arm will then be stable due to the "simple" impedance controller's stability properties [78, 79].

2.3.7 Our Procedure for Tuning Scalar Weights in the Objective Function

We followed the procedure described in this section to empirically tune the scalar weights, α_i, of eq. (12) on page 19 in the software simulation testbed, described in section 3.1.1 on page 25. We then found the same values to have acceptable performance on the real robots. As described in section 2.3.4 on page 19, we set $\alpha_1 = 1$, and tuned the other weights relative to it.

The second term of the objective function (section 2.3.5.1 on the previous page) places a penalty on large changes in the applied joint torques. Increasing the scalar weight associated with this term, α_2, discourages all motions of the arm. We use this term to prevent large motions in the redundant degrees of freedom of the robot. We set $\alpha_2 = 0.00001$. With a value significantly smaller than 0.00001, we observed large motions of the redundant degrees of freedom of the simulated robot. With a value larger than 0.00001, the simulated robot moved more slowly than the commanded end effector velocity in free space, suggesting that α_2 was larger than it needed to be.

For the third term of the objective function, described in section 2.3.5.2 on the preceding page, we tuned α_3 and $\Delta f_{c_i}^d$ together. A large value for $\left\|\Delta f_{c_i}^d\right\|$ helps to decrease the force quickly but can

also cause the arm to unecessarily "bounce away" from contacts or break contact due to modeling errors.

A large value for α_3 causes the controller to place a higher priority on decreasing the contact forces compared to moving towards the goal, which can cause sluggish motion of the arm in situations where it needs to slide along an object with the contact force hovering around the *don't care force threshold*. A large α_3 results in the controller reducing the contact force (at the expense of sliding along the object) the moment it goes above the *don't care force threshold*. Small values for α_3 and $\left\|\Delta f_{c_i}^d\right\|$ causes the contact force to decrease very slowly once it goes above the *don't care force threshold*. For this work, we set the scalar weight $\alpha_3 = 1$ and $\left\|\Delta f_{c_i}^d\right\| = 0.2N$.

2.3.7.1 Units of the Different Terms in the Objective Function

We would also like to point out that without the scalar weights, the three quadratic terms in the objective function have different units:

1. $g_1 = \|\Delta x_d - \Delta x_h\|^2$ (eq. (13) on page 19) has the unit m^2 and dimension L^2
2. $g_2 = \|\Delta\tau\|^2$ (eq. (21) on page 21) has the unit $\mathrm{N}^2\mathrm{m}^2$ and dimension $\mathrm{M}^2\mathrm{L}^4\mathrm{T}^{-4}$
3. $g_3 = \sum_i \left\|\Delta f_{c_i}^d - \Delta f_{c_i}\right\|^2$ (eq. (22) on page 21) has the unit N^2 and dimension $\mathrm{M}^2\mathrm{L}^2\mathrm{T}^{-4}$

The scalar weights α_1, α_2, and α_3 can be interpreted as also converting the different terms of the objective function to the same physical dimension so that the summation in the overall objective function, $\sum_{i=1}^{3} \alpha_i g_i$, is physically meaningful. However, we currently do not have an intuitive physical interpretation for the different scalar weights.

2.4 Conclusion

In this chapter, we have presented our approach to manipulation, which from the outset emphasizes the benefits of making contact with the world. We have also presented a new controller and evaluated it with respect to the task of reaching in high clutter. In the next chapter, we present our evaluation of this controller in three distinct testbeds, a software simulation testbed, a real robot arm with simulated tactile sensing, and a real robot with real tactile sensors on the forearm.

CHAPTER III

REACHING IN CLUTTER WITH WHOLE-ARM TACTILE SENSING: EXPERIMENTAL TESTBEDS AND RESULTS

In the previous chapter (Chapter 2 on page 7), we presented our approach to robot manipulation, which emphasizes the benefits of making contact with the world across the entire manipulator. We assumed that the robot to be controlled has low-stiffness actuation at its joints, and tactile sensing across the entire surface of its manipulator. We then presented a novel controller that exploits these assumptions. The controller uses model predictive control (MPC) with a time horizon of length one, and a linear quasi-static mechanical model that it constructs at each time step.

In this chapter, we show that this controller enables both real and simulated robots to reach goal locations in high clutter with low contact forces. While doing so, these robots push aside movable objects, deform compliant objects, and perceive the world through contact. Our tests include a real robot reaching into a cinder block and simulated foliage using a novel tactile sensor array on its forearm. In simulation, we also show that our controller performs better with whole-arm tactile sensing than with a force-torque sensor at each link, and that the relative value of whole-arm tactile sensing increases with higher clutter.

The rest of this chapter is organized as follows. We describe three testbeds we used to evaluate our controller (section 3.1 on the next page). We then describe the approaches that we used for comparison (section 3.2 on page 30), and the experiments that we ran in simulation section 3.3 on page 31 and on a real robot section 3.4 on page 38). We discuss some limitations in section 3.5 on page 44 and conclude in section 3.6 on page 45.

3.1 Experimental Testbeds

We evaluated our model predictive controller, described in the previous chapter (chapter 2 on page 7), using three different testbeds: 1) a software simulation testbed with a 3 DoF planar arm, 2) a hardware-in-the-loop testbed that simulates whole-arm tactile sensing for a real 7 DoF arm, and 3) a real 7 DoF arm with a real tactile sensor covering its forearm. The same MPC code written in Python runs on all three experimental testbeds. For visualization, we use the `rviz` program, which is part of the Robot Operating System [170].

3.1.1 The Simulated Robot

This testbed allows us to simulate a large number of trials. We use the open source physics simulation library Open Dynamics Engine [198] to simulate a planar arm with three rotational joints, a 1kHz joint-space impedance controller, and tactile sensors covering the entire surface of the arm with 100 taxels per meter. Figure 7 shows a visualization of the simulated robot and taxels. Marc Killpack is the lead developer of this testbed.

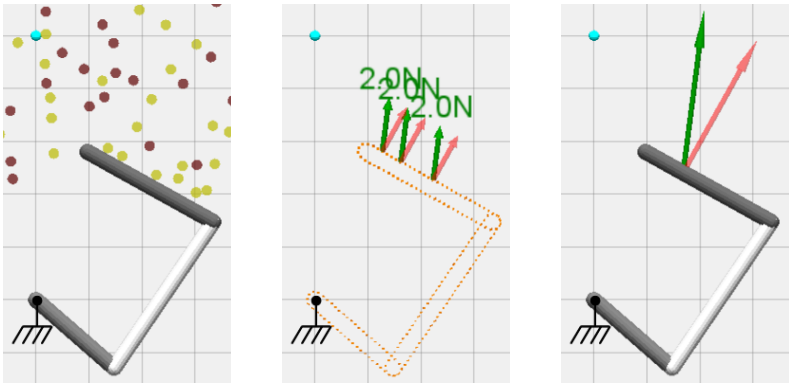


Figure 7: ***Left:*** *Visualization of the three link planar arm reaching to a goal location (cyan) in a volume consisting of rigid cylinders that are either fixed (red) or movable (yellow). The base of the arm is rigidly fixed to the world.* ***Middle:*** *Visualization of the whole-arm tactile sensor. The orange points are 1cm apart and represent the centers of the simulated taxels. The green arrows are the contact force vectors and each red arrow is the sensed component of the contact force normal to the surface of the arm.* ***Right:*** *Visualization of the estimated resultant contact force and contact location sensed with a simulated force-torque sensor at the base of each link (section 3.2.2.1 on page 30). The green arrow is the estimated resultant contact force vector and the red arrow is the normal component of the estimated resultant contact force.*

The simulated three link planar arm has kinematics and joint limits similar to a human holding his hand straight out and manipulating in a plane parallel to the ground at shoulder height. The lengths (19.6cm, 33.4cm, and 28.8cm) and masses (2.8Kg, 2.3Kg, and 1.32Kg) of the different links are similar to a human torso and arm [69]. The three joints, with stiffnesses of 30, 20, and

15 Nm/rad, correspond to the torso, shoulder, and elbow. The joint stiffness values are similar to measured stiffnesses of humans during planar reaching motions [194].

3.1.2 The Real Robot, Cody

We used the robot Cody, first described section 1.2 on page 3, for experiments in this chapter. In this work we control the omnidirectional mobile base with a PID controller that uses visual odometry as we describe in [111]. The passive elastic elements of the SEAs the the arm are stiff relative to the active compliance we specify for all joints except two wrist joints. The passive elastic elements help protect the actuators from shock loads ([167]) and may have advantages in wedging and jamming conditions in high stiffness clutter, but we do not investigate their role in this paper. For the experiments in this chapter, except for two wrist joints, the actuators can be thought of as torque sources.

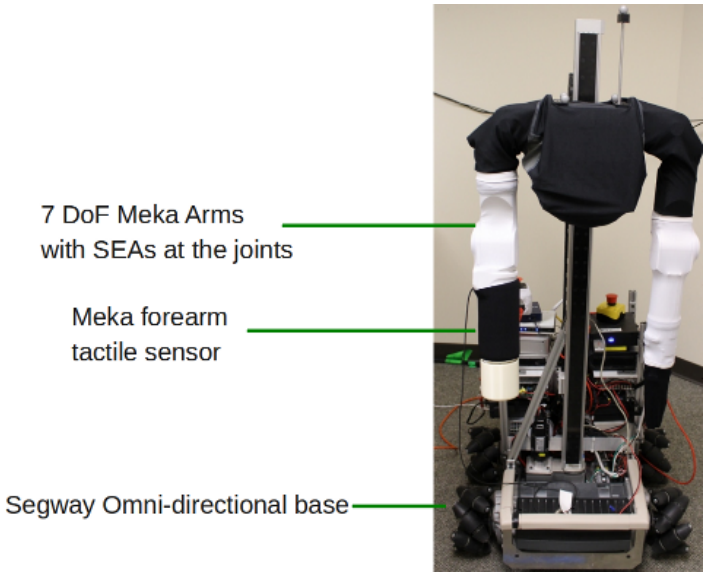


Figure 8: *The robot Cody with compliant 7 DoF arms and a tactile sensor on its right forearm.*

On Cody, "simple" impedance control runs at 1kHz within an RTAI real-time thread on a computer running Ubuntu Linux. We use the same stiffness settings as our work on door opening (chapter 5 on page 63 and [92, 94]). We set the stiffnesses for the shoulder flexion/extension, shoulder abduction/adduction, shoulder internal/external rotation, elbow flexion/extension, and forearm pronation/supination motions at 20, 50, 15, 25, and 2.5 Nm/rad, respectively. The joint stiffness values are also similar to measured stiffnesses of humans during planar reaching motions [194]. We use position control for the wrist flexion/extension and abduction/adduction, since the passive compliance of the SEA springs and the cables connecting the SEAs to these two joints introduce

significant compliance.

3.1.3 Hardware-in-the-loop Simulation of Tactile Sensing

Since we currently do not have whole-arm tactile sensing on a real robot, we have built a hardware-in-the-loop testbed to simulate whole-arm tactile sensing. Figure 9 shows the current implementation of this testbed. The clutter field consist of rigidly mounted posts made of 80/20 aluminum covered with bubble wrap to make them cylindrical and deformable. These posts are similar to the fixed cylinders in software simulation.

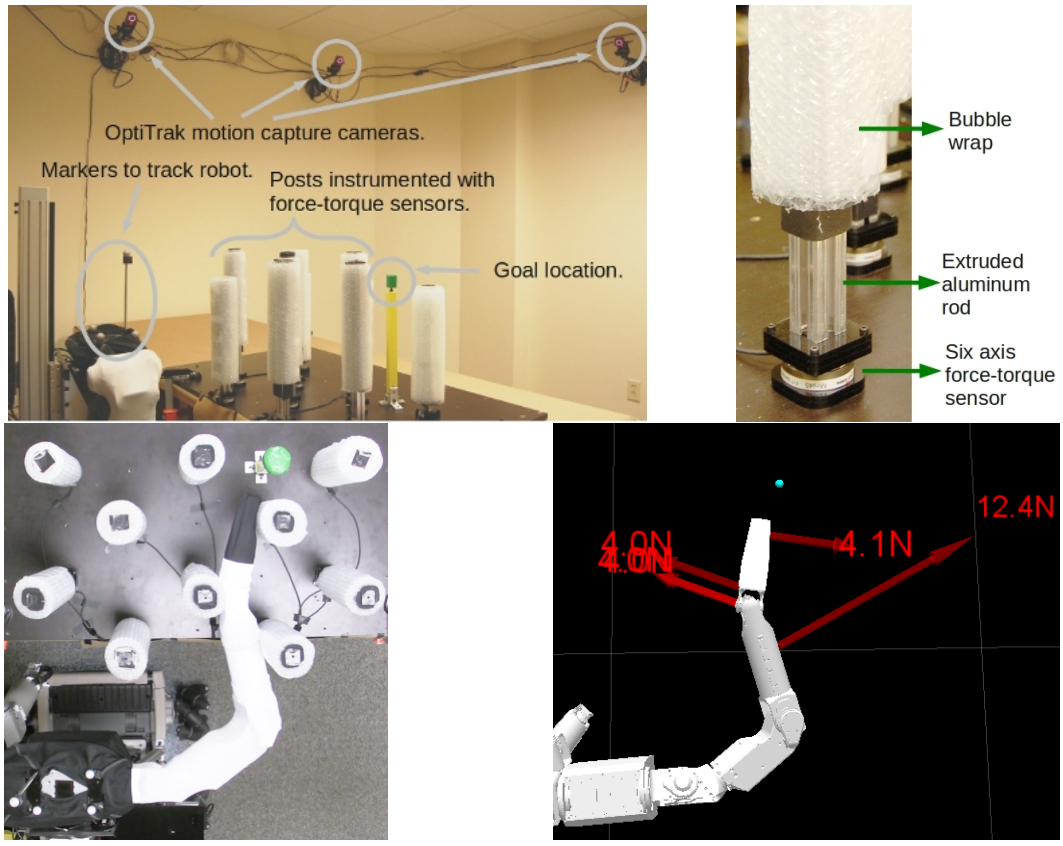


Figure 9: ***Top Left:*** *Different components of the hardware-in-the-loop testbed.* ***Top Right:*** *Close-up of one instrumented obstacle showing the force-torque sensor at the base of an extruded aluminum rod which we have covered in bubble wrap.* ***Bottom Left:*** *Cody attempting to reach to a goal location (green).* ***Bottom Right:*** *Visualization of the simulated tactile sensing.*

The testbed estimates contact locations on the robot's arm using geometric collision detection from OpenRAVE [43] and models of the robot arm and the posts. We use an OptiTrak motion tracking system to estimate the positions of the posts, and the pose of the robot with respect to a common frame of reference.

To estimate contact forces, we instrumented each of the posts with a six-axis force-torque sensor

(ATI Mini45) at its base. For each instrumented post and robot link in contact with one another, the simulated tactile sensor produces a single contact location, which is the centroid of the estimated contact region, and an associated force vector from the post's force-torque sensor. If multiple links make contact with the same post, it divides the force magnitude equally among them.

3.1.4 Real Tactile Sensing Forearm

Figure 10 shows the tactile sensor that covers the forearm of the robot Cody. Meka Robotics and the Georgia Tech Healthcare Robotics Lab developed the forearm tactile sensor, which is based on Stanford's capacitive sensing technology, as described in [37].

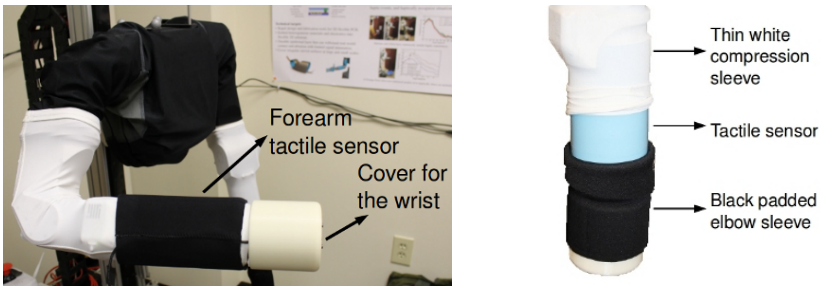


Figure 10: ***Left:*** *Tactile sensor on the right forearm of Cody (underneath the black neoprene sleeve) as well as a 3D printed cover for the wrist.* ***Right:*** *Two layers of material (thin white compression sleeve and black padded sleeve) that we added on top of the tactile sensor (blue).*

The forearm sensor consists of 384 taxels arranged in a 16$\times$ 24 array. There are 16 taxels along the length of the cylindrical forearm and 24 taxels along the circumference. Each taxel has a dimension of 9mm $\times$ 9mm and a sensing range of 0-30N. We can obtain the 16$\times$ 24 taxel array sensor data at 100 Hz using Robot Operating System (ROS) drivers.

On the robot Cody, we added two layers of material on top of the forearm sensor to cover the open parts of the joints, protect the sensor, and make the exterior of the arm low friction. These are shown on the right in fig. 10. The white sleeve is a thin neoprene McDavid compression arm sleeve (Model No. 656T), and the black layer is a padded Ergodyne neoprene elbow sleeve (Model No. 650) designed for athletes.

When using the real tactile sensor, we mounted a rigid 3D printed cylindrical cover over the wrist of the robot, also shown in fig. 10. This cover serves to detect contact forces across the entire surface of the most distal part of the arm using the wrist-mounted force-torque sensor. It also creates a smooth surface over the wrist joint. The cover is flush with the tactile sensing forearm such that the last two wrist joints are unable to rotate when it is mounted.

3.1.5 Detecting Contact

As mentioned section 2.3.3 on page 16, our model predictive controller uses the measured force to detect a contact. Specifically, the controller only adds a contact to its model if the force magnitude is greater than a threshold.

This threshold is 0.5N for the simulated robot and 1N in the hardware-in-the-loop testbed. For the real tactile sensor, we precompute the standard deviation of the sensor noise for each taxel when the arm is not in contact with the environment. The threshold for detecting contact is 6 times the standard deviation of the sensor noise or 1N, whichever is greater. For contacts on the wrist cover, this threshold is again 1N.

3.1.6 Tactile Sensing Limitations

Due to limitations in our implementations, there are differences in tactile sensing with the three testbeds.

The tactile sensors for the software simulated robot and the real tactile sensing forearm match the assumptions we used in our controller derivation. They report the geometric centers of the taxels at which contact is detected along with the total normal force measured at each taxel.

While testing the real tactile sensing forearm, the plastic cover on the distal part of Cody's arm also reports contact forces. These forces are not handled in the same way as our derivation. Instead, the contact location is the center of the force-torque sensor, and the contact force is the total resultant force measured by the force-torque sensor. The controller treats this force vector as though it were a normal force applied at the center of the force-torque sensor, which implies that it will not ignore frictional forces. In other words, the spring associated with this contact in the mechanical model is oriented along the direction of this force vector rather than a surface normal.

Similarly, the hardware-in-the-loop simulation of tactile sensing does not report normal forces, and instead provides the centroid of the estimated contact region as the contact location, and a scaled version of the total resultant force vector measured by the object's force-torque sensor as the contact force.

The primary consequence of these differences is that frictional forces will influence the controller rather than being ignored. This influence is mitigated by the smooth, low-friction surfaces of the plastic cover and Cody's arm. We put neoprene McDavid compression arm sleeves on Cody's arm when it operates in the hardware-in-the-loop system.

3.2 Approaches used for Comparison

In this section, we describe four methods against which we compared our controller: 1) a baseline controller; 2) our model predictive controller with force-torque sensors; 3) our model predictive controller without force sensing, and 4) a state of the art geometric motion planner that has full knowledge of the environment.

3.2.1 Baseline Controller

Our baseline controller uses the same joint-space impedance control as our model predictive controller to maintain low stiffness at the joints. However, it does not use feedback from the tactile sensor, except to define a safety stopping criterion. Specifically, this controller computes

$$\Delta\phi^* = \left(J_h^T J_h\right)^{-1} J_h^T \Delta x_d, \tag{24}$$

where $\Delta\phi^* \in \Re^m$ is the incremental change in the joint-space virtual trajectory (see fig. 5 on page 15), $J_h \in \Re^{3\times m}$ is the Jacobian at the robot's end effector (hand), and $\Delta x_d \in \Re^3$ is the desired Cartesian motion of the end effector computed from eq. (14) on page 20. The baseline controller monitors the tactile sensor values and stops if the force at any contact goes above the safety force threshold, $f_{c_i}^{safety}$.

If we ignore the constraints, use only g_1 (see eq. (13) on page 19) as the objective function, and the arm is not in contact with the world, then eq. (24) on this page is the solution to the quadratic program for our model predictive controller. In free-space both controllers will attempt to move the end effector along a straight line to the goal with identical low stiffness settings at the joints.

3.2.2 Our Model Predictive Controller with Different Sensors

To compare whole-arm tactile sensing with other sensors, we simulated an arm with force-torque sensors at the base of each link and an arm with no force sensors.

3.2.2.1 Simulated Force-Torque Sensors

We used simulated force-torque sensors to calculate the resultant force on each link and its line of action using Poinsot's theorem [144]. We estimated the contact location as the intersection of the line of action of the resultant force with the link axis. This enabled us to estimate a single contact force and contact location for each link. Figure 7 on page 25 shows a visualization of the feedback from the simulated force-torque sensors.

These simulated force-torque sensors are idealized, and hence give an upper bound on performance. The sensors do not have any noise or drift, and they only provide forces due to contact with the environment.

3.2.2.2 No Force Sensing

The arm with no force sensing performs "simple" impedance control, but provides no contact locations nor contact forces to our model predictive controller. Consequently, the controller believes it is operating in free space and always changes the virtual trajectory so as to best move the end effector towards the goal, subject to constraints on the joint limits, joint torques, and such. Although it does not use force sensing for maneuvering, if a contact force exceeds the safety threshold, our model predictive controller is stopped as described section 2.3.6 on page 22.

3.2.3 Motion Planner with Full Knowledge

For experiments in the software simulation testbed, we also compare against a bi-directional RRT motion planner (Bi-RRT) as implemented in OpenRAVE [43]. The motion planner has complete knowledge of the fixed cylinders in the cluttered environment and we remove all the movable cylinders. We use Bi-RRT to estimate whether or not a solution exists for a given goal location and configuration of the fixed cylinders. We use this to estimate an upper bound for the success rate for a given set of trials. This is an approximate upper bound, since there may be no way to reach the goal when the movable cylinders are present.

3.3 Software Simulation Experiments

In this section, we describe experiments with the software simulation testbed (see section 3.1.1 on page 25). We have posted code, data, and instructions to reproduce these results, as described in appendix A.1 on page 147.

3.3.1 Simulating Clutter

For all experiments, the simulated arm reaches to a goal location in a clutter field. Clutter fields are composed of rigid cylinders with a diameter of 2cm that are either fixed to the ground or movable. In isolation, a movable cylinder slides with the application of approximately 2N (static friction and kinetic friction are set to be equal).

To generate a clutter field, we randomly place a predefined number of fixed and movable cylinders into a 1.2m $\times$ 0.6m rectangular region. For each cylinder, we uniformly sample a 2-dimensional

location from the rectangular region, redrawing until we find a location for which the associated cylinder does not collide with the cylinders that have already been placed.

We use this simplified model of clutter to randomly generate clutter fields with different ratios of fixed to movable clutter, and different densities of clutter. This clutter model shares some similarities with dense foliage, which often consists of stiff branches and movable leaves. Reaching into this type of clutter results in a diverse set of contact conditions with numerous point contacts occurring across the arm, some of which result in high forces when pushed against and others of which result in lower forces. Making contact with the clutter also results in secondary effects due to contact between the cylinders.

3.3.2 Regulating Contact Forces

To test the influence of the *don't care force threshold*, $f_{c_i}^{thresh}$, we generated 25 clutter fields, each with 20 fixed and 20 movable cylinders and eight constant goal locations as shown in fig. 12 on the next page. We then ran our model predictive controller with five different *don't care force threshold* values on these 200 trials.

Figure 11 on this page shows the 75^{th}, 95^{th}, and 99^{th} percentile scores for the magnitude of the contact force for each value of $f_{c_i}^{thresh}$. The correlation coefficient between $f_{c_i}^{thresh}$ and the 95^{th} percentile score was ≥ 0.999, providing evidence that the $f_{c_i}^{thresh}$ parameter can be used to predictably influence the contact forces. For the contact force statistics we report here and in other simulation experiments, we use the raw contact forces from the physics simulation, rather than the contact forces provided by various sensor configurations (e.g., normal forces) and used by our model predictive controller.

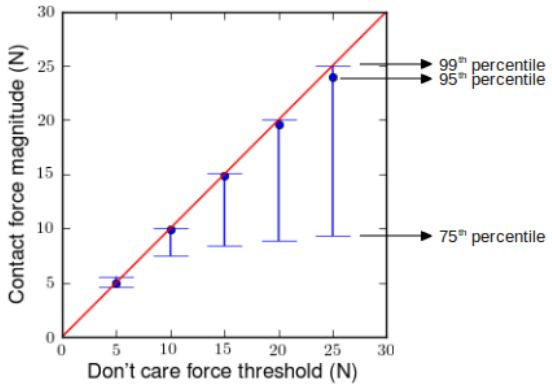
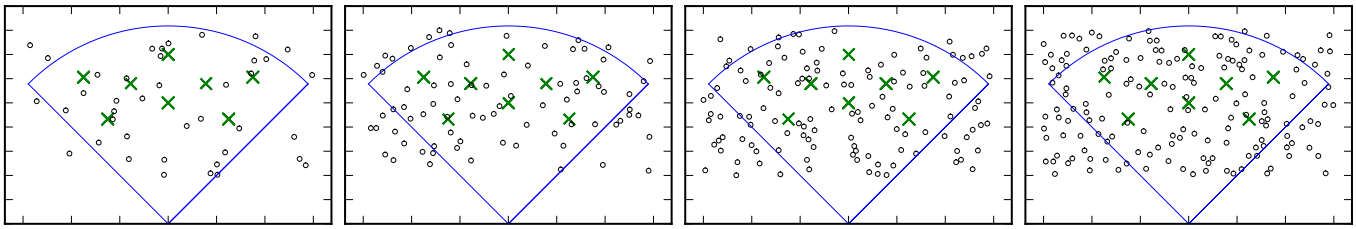


Figure 11: *75^{th}, 95^{th}, and 99^{th} percentile scores for the contact force from 200 trials of single reaches to eight different goal locations in 25 distinct clutter fields for different values of the don't care force threshold. Details are in section 3.3.2 on this page.*



(a) 40 cylinders (b) 80 cylinders (c) 120 cylinders (d) 160 cylinders

Figure 12: *An example each of the four different levels of clutter that we used in the evaluation of section 3.3.3. The cylinders are the black circles, the eight goal locations are green crosses, and the blue curve outlines the area swept by the fully extended arm when its proximal joint moves from 45°to 135°. The physical limits for the proximal joint are at -150°and 150°.*

3.3.3 21600 Reaching Trials

We used the simulation testbed to compare our model predictive controller with whole-arm tactile sensing to a version with force-torque sensors, a version with no force sensors, and Bi-RRT, as described in section 3.2.2 on page 30.

We varied the level of clutter by changing the total number of cylinders from 40 to 160 in increments of 40. We also changed the composition of the clutter to have movable cylinders constitute 25%, 50%, or 75% of the clutter with the rest fixed. Overall, we ran $3 \times 4 \times 3 \times 75 \times 8 = 21600$ trials of the simulated robot reaching in clutter with three types of sensor feedback, four levels of clutter, three ratios of movable to fixed cylinders, seventy-five randomly generated clutter field configurations, and eight constant goal locations. Figure 12 shows examples of the different levels of clutter and the goal locations.

For all trials, we set $f_{c_i}^{thresh} = 5N$ and $f_{c_i}^{safety} = 50N$ for each contact c_i.

3.3.3.1 Success Percents for Three Controllers and Bi-RRT

Figure 13 on the following page shows the performance of the three controllers, and our estimated upper bound for the success percent via Bi-RRT, which we will refer to as the estimated optimal success percent. As expected, the estimated optimal success percent and the controllers' success percents decrease with increasing clutter.

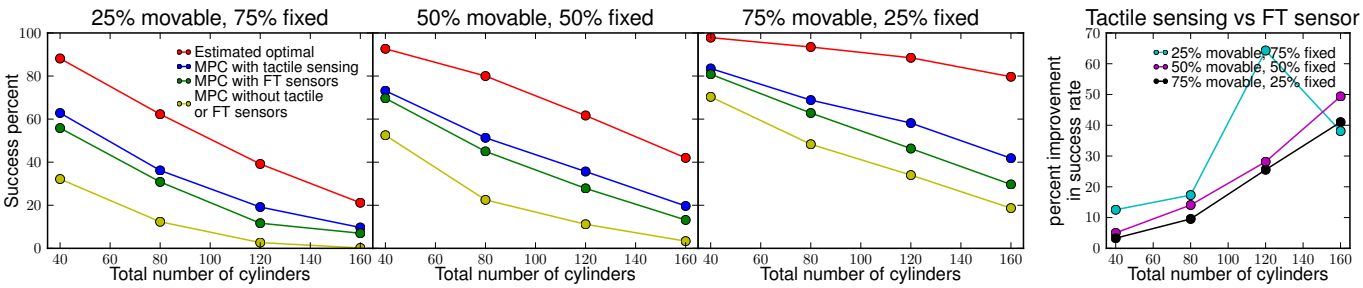


Figure 13: ***Left:*** *Success percent for a single reach attempt in different levels of clutter and ratios of fixed to movable clutter.* ***Right:*** *Percent improvement in the success rate when the robot uses whole-arm tactile sensing instead of force-torque sensors. Details are in section 3.3.3.1.*

Our model predictive controller with whole-arm tactile sensing has a higher success percent compared to the same controller with force-torque sensors for all levels of clutter and ratios of fixed to movable cylinders. The relative value of whole-arm tactile sensing also increases as the level of clutter increases, with one exception. For example, with an equal ratio of fixed and movable cylinders, this percent improvement is around 5% with 40 cylinders and 50% with 160 cylinders. There is one data point that goes against this trend. The percent improvement with 25% movable and 75% fixed and 120 total cylinders is higher than other trials. The cause of this result is unclear.

3.3.3.2 Examples of Whole-arm Tactile Sensing Succeeding when Force-Torque Sensing Fails

Figure 14 on the next page shows two examples of multi-contact situations where force-torque sensors failed, but whole-body tactile sensors succeeded. In the example on the left, the robot with force-torque sensors became stuck due to contacts on both sides of the same link. In the example on the right, the robot with force-torque sensors became stuck due to multiple contacts with low forces that it interpreted a single high force contact.

We provide links to videos of these two trials in appendix A.1 on page 147. In the videos, the robot starts from the same initial configuration in free space, fails with force-torque sensors and succeeds with whole-arm tactile sensors. Additionally, for these two trials, we first ran the controller with force-torque sensing until it became stuck, and then switched to whole-arm tactile sensing. The robot was then able to successfully reach the goal starting from the state at which it had been stuck.

3.3.3.3 Contact Forces for Three Controllers

Table 1 on the next page lists the average values of the maximum force that occurred in each of the trials, including both successful and unsuccessful attempts to reach to the goal. As noted previously, these contact force statistics use the raw contact forces from the physics simulation.

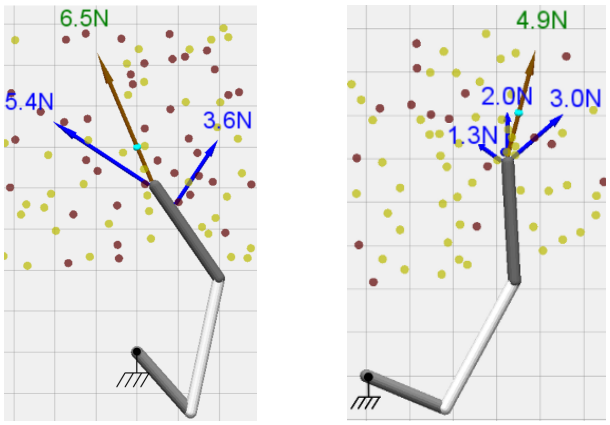


Figure 14: *Two multi-contact situations from the trials of section 3.3.3 on page 33 in which our model predictive controller gets stuck with feedback from force-torque sensors and succeeds with whole-arm tactile sensing. The cyan circle is the goal location, the blue arrows are the forces at the individual contacts, and the brown arrow is the resultant contact force and contact location, estimated using the force-torque sensor.* ***Left:*** *The distal link of the robot makes contact with two fixed cylinders.* ***Right:*** *The distal link is in contact with three movable cylinders. Each individual contact force is low but the resultant is equal to* f^{thresh}.

The average maximum force is higher with force-torque sensing than whole-arm tactile sensing and this difference is larger for higher clutter. Our model predictive controller with force-torque sensing is overly conservative in situations where multiple contact forces on the same link point in similar directions, and underestimates the contact force if there are contact forces that cancel each other.

Table 1: *Average of the maximum contact force over 600 trials in varying levels of clutter.*

	Total number of cylinders			
	40	80	120	160
25% movable, 75% fixed				
MPC with tactile sensing	5.3N	6.7N	8.9N	10.2N
MPC with force-torque sensors	5.6N	7.4N	9.1N	12.1N
MPC without tactile or FT sensors	33.3N	39.1N	40.6N	43.2N
50% movable, 50% fixed				
MPC with tactile sensing	4.6N	6.2N	8.0N	9.8N
MPC with force-torque sensors	4.8N	6.6N	8.5N	14.5N
MPC without tactile or FT sensors	25.6N	34.2N	39.1N	42.7N
75% movable, 25% fixed				
MPC with tactile sensing	4.1N	5.6N	6.8N	8.5N
MPC with force-torque sensors	3.9N	7.3N	8.5N	11.7N
MPC without tactile or FT sensors	17.0N	25.9N	30.5N	37.7N

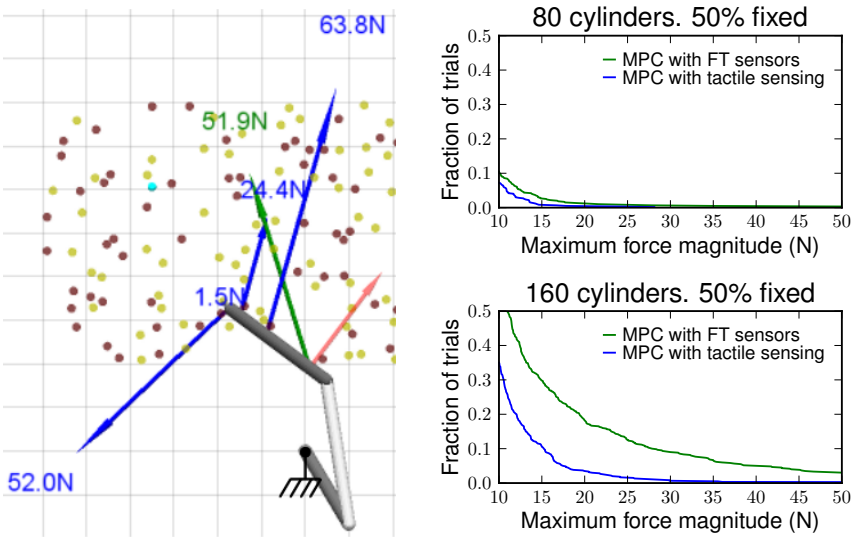


Figure 15: ***Left:*** *An example where our model predictive controller with force-torque sensors resulted in the robot applying large forces. The blue arrows are the actual contact forces. The green arrow is the resultant force vector and the red arrow is its component normal to the surface of the arm.* ***Right:*** *Fraction of 600 trials each with 80 and 160 cylinders (50% fixed, 50% movable) for which the maximum magnitude of the force applied to the environment was greater than the value on the x-axis.*

Figure 15 shows in more detail that the number of occurrences of high force is greater with force-torque sensing compared to whole-arm tactile sensing, and that this difference is greater with higher clutter.

3.3.3.4 Average Velocity for Three Controllers

1.27 cm/s, 1.44 cm/s, and 1.66 cm/s were the average end effector velocities for our model predictive controller with whole-arm tactile sensing, with force-torque sensing, and without force sensing, respectively. We have not optimized these controllers for speed. The velocities include both successful and failed trials. A number of factors could be responsible for the slower average velocity with tactile sensing, including the complexity of the trials for which it was successful, and the other two controllers were not. Likewise, without force sensing, the controller is free to apply higher forces and more quickly push through clutter.

3.3.3.5 Local Minima

Figure 16 on the following page shows an example of a local minimum in which our controller can become stuck. To escape from local minima, some form of high-level control is required. For some of our experiments, we used a high-level controller that attempts to move the end effector in

a straight line towards the goal, detects when the arm has stopped making progress (reached a local minimum), pulls the arm back to a new configuration, and then tries again.

We provide links to a video showing this additional control layer in a simplified environment in appendix A.1 on page 147.

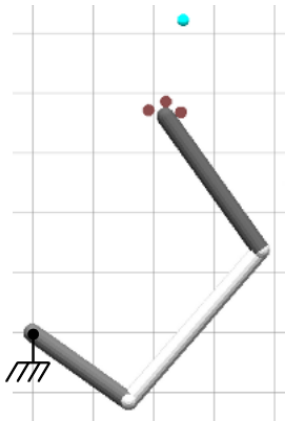


Figure 16: *An example of our greedy model predictive controller stuck in a local minimum. The arm can reach the goal location (cyan circle) if it goes around the cylinders (red circles) but this would involve an increase in the cost function.*

Using this additional control layer, we tested our controller with whole-arm tactile sensing. We allowed our greedy controller to pull out and restart from up to five different end effector positions equally spaced along a line in front of the clutter field rectangle. Table 2 on the following page shows the success percent when we allowed our model predictive controller to retry up to four times (five total reach attempts).

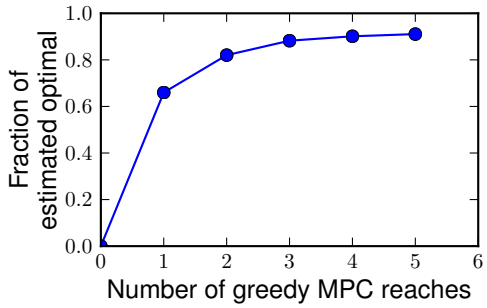


Figure 17: *Success rate of our model predictive controller with whole-arm tactile sensing as a fraction of the estimated optimal success rate.*

Figure 17 shows how the success rate as a percentage of the estimated optimal success rate increases as the number of reach attempts increases. With a single reach, the success rate of our model predictive controller with tactile sensing was approximately 66% of the estimated optimal. With up to five reaches, the success rate was around 91% of the estimated optimal. This success

rate is across all three ratios of fixed to movable clutter and four levels of clutter.

Table 2: *Success rate with multiple reaches with whole-arm tactile sensing in varying levels of clutter.*

	Total number of cylinders			
	40	80	120	160
25% movable, 75% fixed				
Single reach	62.8%	36.2%	19.2%	9.7%
Up to 5 reaches	82.7%	59.7%	35.2%	17.5%
Estimated optimal	88.1%	62.3%	39.2%	21.2%
50% movable, 50% fixed				
Single reach	73.2%	51.3%	35.7%	19.7%
Up to 5 reaches	89.0%	72.0%	54.8%	36.7%
Estimated optimal	92.7%	80.0%	61.2%	42.0%
75% movable, 25% fixed				
Single reach	83.5%	68.3%	58.2%	41.8%
Up to 5 reaches	94.2%	87.3%	77.2%	64.5%
Estimated optimal	97.8%	93.5%	88.5%	79.7%

3.4 Experiments with a Real Robot

In this section, we present results from experiments with the real robot named Cody.

3.4.1 Hardware-in-the-loop Tactile Sensing Simulation Testbed

The experiments in this section use the hardware-in-the-loop testbed to simulate whole-arm tactile sensing.

3.4.1.1 Selective Control of Force Applied to Different Regions of the Environment

With this experiment, we illustrate that our model predictive controller can be used to selectively control the contact forces in different regions. We defined a 'fragile' cylindrical volume with respect to the world frame. If the location of a contact c_i in the world frame was within the 'fragile' volume, we set the *don't care force threshold*, $f_{c_i}^{thresh}$, to $2N$. For contact outside this volume, we set the *don't care force threshold* to $5N$.

Figure 18 on the next page shows histograms of the contact forces while reaching to the goal location. The contact forces within the 'fragile' region had a median of $1.76N$, while the contact forces outside of the 'fragile' region had a median of $4.54N$.

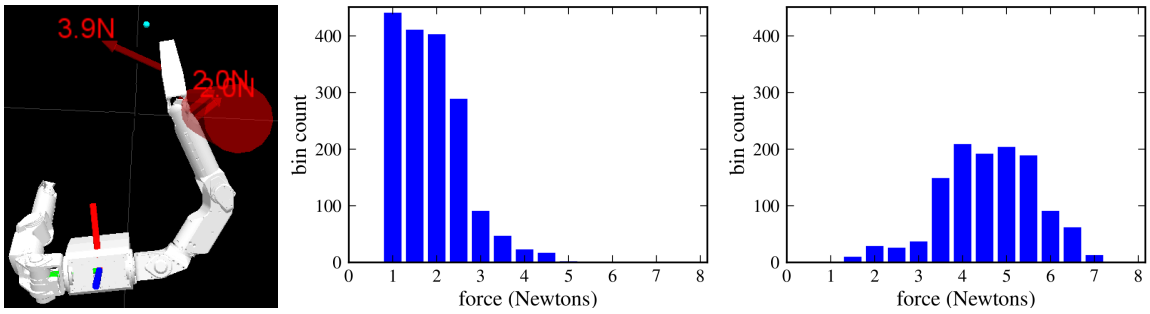


Figure 18: *Experiment to demonstrate selective control of contact force in different regions using our model predictive controller.* ***Left:*** *Tactile sensing visualization when some contacts are within the 'fragile' region (red cylinder) and other contacts are outside the 'fragile' region. The cyan circle is the goal location that the robot successfully reached.* ***Middle:*** *Histogram of contact forces within the 'fragile' region.* ***Right:*** *Histogram of contact forces outside the 'fragile' region.*

3.4.1.2 Test of Our Model Predictive Controller as Part of a Fully Autonomous System

Using the hardware-in-the-loop testbed, we tested our model predictive controller as part of a fully autonomous system. The autonomous system had two high-level controllers. The first high-level controller (HC1) moved the robot's mobile base to one of three pre-defined positions equally spaced along a line in front of the clutter field. The robot would then hold its arm in a pre-defined pose while slowly moving forward towards the clutter field until it detected contact with its arm.

At this point, the second high-level controller (HC2) took over. HC2 attempted to reach the goal location using our model predictive controller. If HC2 detected that the end effector stopped making progress towards the goal location, it would attempt another reach or pull the arm out and return control to HC1. For each of up to four reach attempts, HC2 would perform some combination of pulling the arm out and moving the end effector to the left or right of a post. In appendix A.1 on page 147, we provide a link to a video that shows the full results of successfully reaching the 5 goal locations shown in fig. 19.

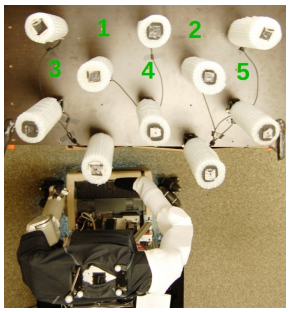


Figure 19: ***Left:*** *Five different goal locations within the hardware-in-the-loop testbed that we used to test our model predictive controller as part of a fully autonomous system.*

We also recorded the contact forces that occurred during the successful reach for each of the 5 trials. The average of the maximum force for each of these 5 reaches was 5.6N. The average of the contact forces in these 5 reaches that exceeded the *don't care force threshold*, $f_{c_i}^{thresh} = 5N$, was 5.5N.

Additionally, for each of the 5 initial conditions (mobile base location and arm pose) from which HC2 succeeded, we attempted a single reach with the baseline controller. The baseline controller succeeded with 3 out of 5 of these initial conditions, had an average maximum contact force of 17.7N, and an average contact force above 5N of 14.3N. This indicates that 2 of the 5 successful reaches performed by the fully autonomous system benefitted from our model predictive controller, HC2, or both. Our controller also kept the contact forces lower than the baseline controller.

3.4.2 Real Tactile Sensing Forearm

We also conducted experiments with the forearm tactile sensor described in section 3.1.4 and fig. 10 on page 28. Since the tactile sensor only covers the forearm of the robot, we designed the experiments to restrict contact to the forearm and the plastic cover on the wrist.

We performed experiments with clutter that simulated foliage and rubble. Our simulated foliage consists of both compliant objects (plastic leaves) and fixed rigid objects (blocks of wood). The leaves can result in substantial occlusion (see fig. 2 on page 8), and can be pushed aside with relatively low force. However, the blocks of wood are hidden, rigid, and effectively immovable.

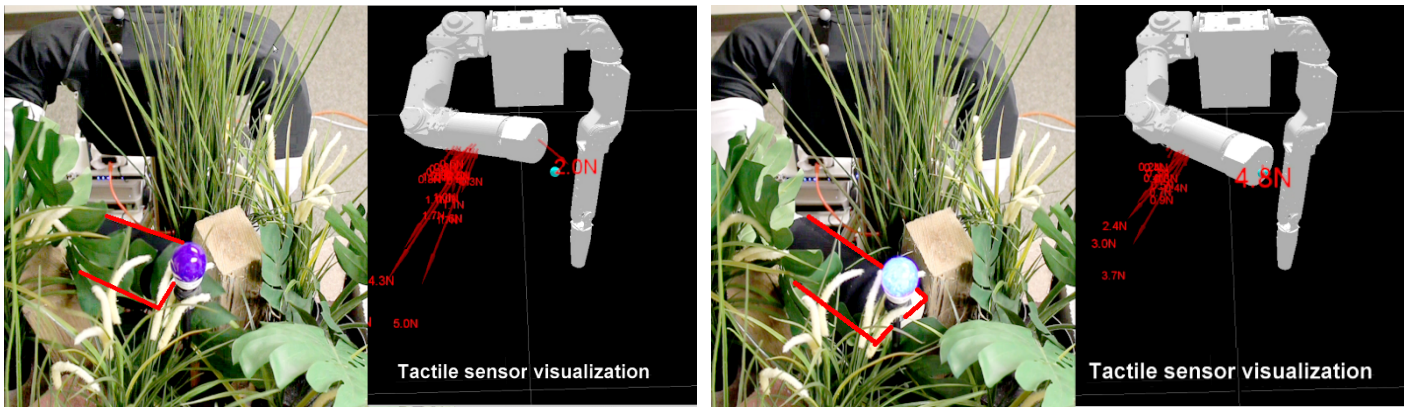
The cinder block is a rigid, heavy, and fixed object, representative of some of the objects a robot would encounter in rubble. The diameter of the robot's forearm is $10cm$. It is close to the width of the opening of the cinder block, which varies between $13.5cm$ and $14.5cm$. Additionally, the edges are sharp and the surface is abrasive.

3.4.2.1 Illustrative Examples – Foliage and Cinder Block

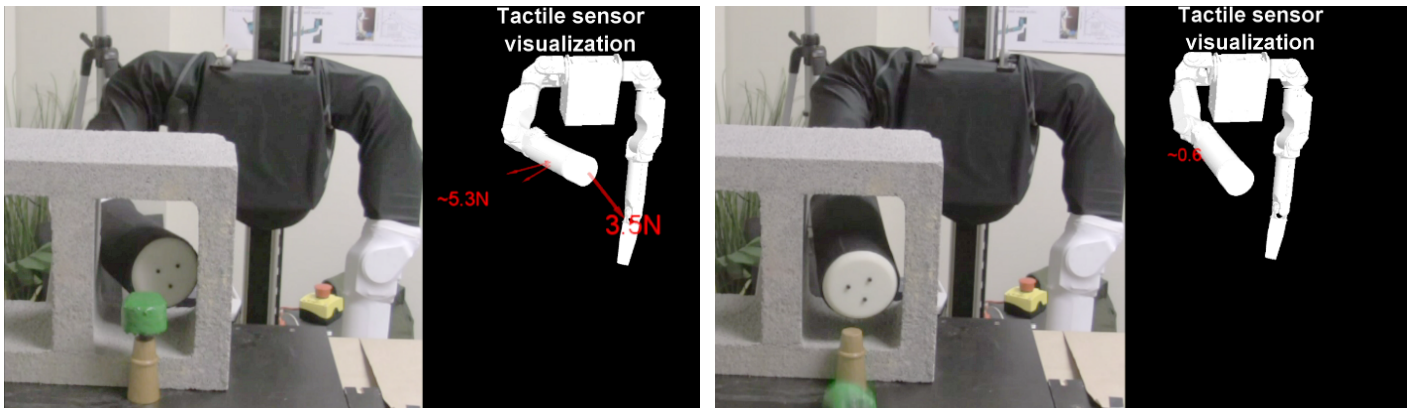
We performed one trial each of the robot reaching to a goal location in foliage and reaching through the opening of a cinder block. fig. 20 on the next page shows two images and the histograms of the contact forces for these two trials. The average end effector velocities were 2.95 cm/s and 2.14 cm/s respectively. We provide links to videos of these two trials in appendix A.1 on page 147.

3.4.2.2 Model Predictive Controller vs Baseline Controller in Foliage

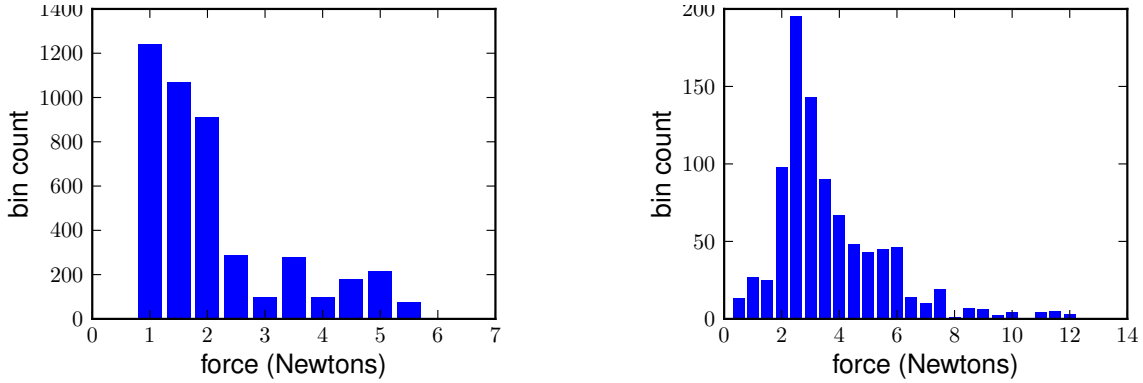
For a more thorough evaluation of our model predictive controller and the baseline controller in realistic conditions, we performed five trials with automatically generated goal locations that were



(a) Reaching to a goal location in foliage with multiple contacts along the arm. The forearm and 3D printed cover for the wrist are approximately outlined in red. The goal location is vertically below the blue bulb, and is the cyan circle in the visualization.



(b) Reaching to a goal location (green) through the opening of a cinder block.



(c) Histogram of contact forces while reaching to a goal location in foliage (left), and through the opening of the cinder block (right).

Figure 20: *Cody reaching to a goal location in realistic conditions using its forearm tactile sensor, described in section 3.4.2.1 on the preceding page.*

equally spaced along a line within our simulated foliage, as shown in fig. 21 on the next page.

We started each trial by positioning the robot at the same location in front of the clutter. The

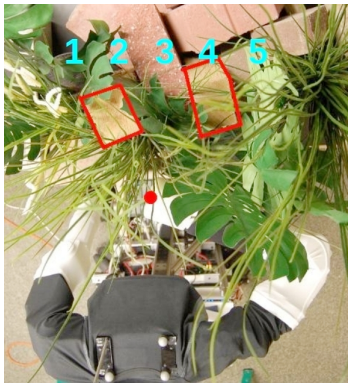


Figure 21: *Five different goal locations in simulated foliage that we used to compare the model predictive controller and the baseline controller, described in section 3.4.2.2 on page 40. The environment consists of compliant leaves and rigid blocks of wood (outlined in red). The red circle denotes the position of the end effector.*

robot then autonomously moved its mobile base to four roughly equally spaced positions along a line, and attempted to reach to the goal location using both the model predictive controller and the baseline controller, as shown in fig. 22.

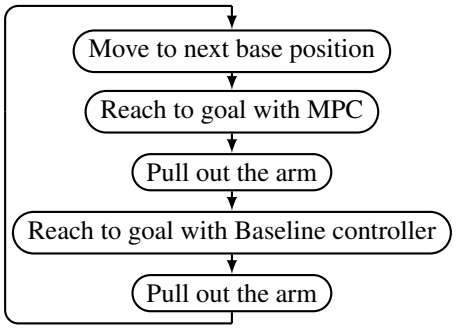


Figure 22: *Different steps that the robot performed for each of the five trials in our model predictive controller vs baseline controller comparison in foliage, described in section 3.4.2.2 on page 40.*

Table 3 on the following page shows the results from the twenty reach attempts (4 base positions $\times$5 trials) for each controller. The model predictive controller successfully reached goal locations 1, 3, and 5, while the baseline controller was only successful for goal location 5. Further, our model predictive controller successfully kept the contact forces around the *don't care force threshold*, $f_{c_i}^{thresh} = 5N$. In contrast, the baseline controller exceeded the safety force threshold, $f_{c_i}^{safety} = 15N$, 19 out of 20 times. We provide a link to a video of the complete experiment in appendix A.1 on page 147.

Table 3: *Model predictive controller vs baseline controller in foliage.*

	MPC	Baseline Controller
Success rate	3/5	1/5
Exceeded safety threshold (15N)	0/20 attempts	19/20 attempts
Avg. max. contact force	5.5N	14.5N
Avg. contact force above $f_{c_i}^{thresh}$ (5N)	5.2N	9.2N

3.4.2.3 Contact Stiffness Parameter's Effect on Performance

In this section, we describe an experiment to investigate how the stiffness used by the controller to model each contact, K_{c_i}, affects the controller's performance. For all other experiments in this chapter, we set this parameter to a high stiffness value. For example, in the software simulation experiments and the other experiments involving the forearm tactile sensor, we set this parameter to be 1000N/m. This is a conservative value that makes the robot think that the world is stiff. Consequently, the robot tends to move slowly while in contact, since it predicts that contact forces will rise quickly for small displacements.

As we show in this section, when the robot makes contact with a low-stiffness object, this high parameter value makes the robot move unnecessarily slowly. With a lower stiffness value that better matches the compliance of the object, the robot pushes more aggressively into the object to reach the goal location.

For the experiment, we fixtured a cylindrical piece of styrofoam in front of the robot in an upright position, and selected a goal location such that the robot's arm could push and bend the cylinder to reach the goal (see fig. 23 on the following page). We estimated the stiffness of the cylinder by hand and found it to be approximately between 100N/m and 300N/m in the region where the robot pushed on the cylinder. Our measured stiffness values varied with the height of contact and how much the cylinder was displaced.

The robot used our model predictive controller to reach the same goal location 5 times for each of 5 contact stiffness values (200, 500, 1000, 3000 and 5000N/m), giving a total of 25 trials. The robot successfully reached the goal location in all 25 trials.

Table 4 on the next page shows the mean and standard deviation of the execution time and the maximum contact force for each trial, as sensed by the forearm tactile sensor. The results show that as the controller's contact stiffness estimate came closer to the actual stiffness, the execution time decreased while the forces continued to remain below the *don't care force threshold*, which

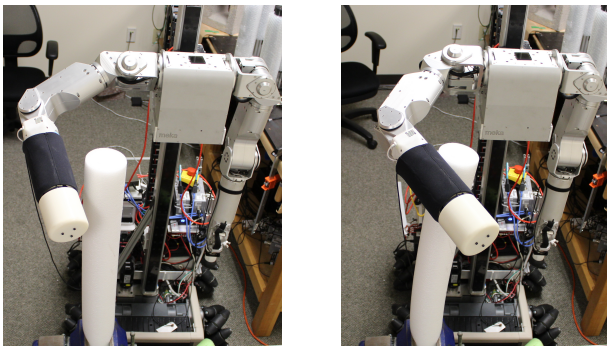


Figure 23: *Initial configuration of the arm **(left)** and a final configuration after it has reached the goal by bending the styrofoam cylinder **(right)**.*

we set to 5N. This suggests that better estimates of contact stiffness could increase the controller's performance. Improved contact stiffness estimates might be gained via online estimation, or by using the statistics of stiffness in a particular type of environment.

Table 4: *Mean (std) over 5 trials of the execution time and maximum force for different values of the modeled contact stiffness while pushing into a compliant object.*

	Modeled contact stiffness (N/m)				
	200	500	1000	3000	5000
Mean (std) execution time	3.44s (0.06s)	3.49s (0.05s)	3.66s (0.04s)	4.49s (0.19s)	5.42s (0.15s)
Mean (std) max force	3.55N (0.17N)	3.87N (0.15N)	3.78N (0.19N)	3.65N (0.17N)	3.69N (0.40N)

3.5 Limitations

We now discuss the limitations of the controller that we described in chapter 2 on page 7 and evaluated in this chapter.

The stability of our controller is unproven. We can, however, make two basic observations. First, as we describe in section 2.3.6 on page 22, when our model predictive controller is stopped due to a contact force that exceeds the safety threshold, the overall controller becomes stable if the environment is passive. Second, if our model predictive controller's quasi-static model were accurate, then the system would achieve static equilibrium in a single time step. However, in addition to various linear approximations, the quasi-static model ignores dynamics and other real-world factors that influence stability. Empirically, we have found that the system behaves in a stable manner, but formal analysis could be beneficial.

Our contact model consists of a linear spring, which is computationally favorable, but predicts

adhesive forces when breaking contact.

Our controller places no penalty on a predicted contact force below $f_{c_i}^{thresh}$, and has a hard inequality constraint that prevents higher predicted forces. It may be advantageous to soften this constraint.

Our controller also ignores dynamics. The resulting quasi-static model is suitable for slow motions, which is reasonable for haptic reaching since a collision could occur at any moment. Taking dynamics into account may result in better performance and higher speeds.

The high-level control we used in this chapter may not be appropriate for some types of clutter. The design of high-level controllers, and associated representations with memory, merits further inquiry. For example, surface following while exploring an environment may be beneficial.

We set the stiffnesses of the joints to constant low values. We did not investigate how performance changes with different values. A related open question is how to initialize and adapt the controller parameters such as the joint stiffnesses, stiffnesses of the springs in the contact model and starting configuration of the arm, given a robot, an environment, and a task.

3.6 Conclusion

In this and the previous chapter, we have presented our approach to manipulation, which from the outset emphasizes the benefits of making contact with the world. We have also presented a new controller and evaluated it with respect to the task of reaching in high clutter. In three distinct testbeds over numerous experiments, our controller with whole-arm tactile sensing outperformed other controllers, including a baseline controller and our controller with force-torque sensing. In simulation, the relative benefits of whole-arm tactile sensing increased as the clutter increased. Moreover, as the number of retries increased, the success rate quickly rose towards our estimate of the optimal success rate. This suggests that detailed models with long time horizon planning may not be necessary when reaching in clutter, which is similar in spirit to research by [31] on bipedal walking. An open question is the extent to which our controller can be used for other tasks, robots, and environments. We expect that controllers designed for whole-arm contact could eventually serve as general purpose controllers upon which other capabilities, such as avoiding contact, are built.

CHAPTER IV

FABRIC TACTILE SENSORS

Robotics research has presented tactile sensor designs for over a decade, but few robots have tactile sensors covering large surfaces of their arms. In this chapter, we present our design for inexpensive, flexible, and stretchable tactile sensor arrays with low spatial resolution using conductive and resistive fabric. Our design is inspired by rSkin, an open source project, and is relatively straightforward to replicate. We investigate some of the challenges involved with covering articulated joints with tactile sensors and present a prototype tactile sensing sleeve that covers the end effector, wrist joints, and forearm of Cody. We also cover large surfaces of a PR2 arm with our fabric tactile sensors.

With an informal experiment, we show that our model predictive controller of chapter 2 and this tactile sensor have the potential to enable robots to manipulate in close proximity to, and in contact with humans while keeping the contact forces low. We also present a single trial of manipulating an object on a shelf with multiple contacts between the robot arm and the environment. The work presented in this chapter is in collaboration with Sarvagya Vaish and Prof. Charles C. Kemp.

4.1 Motivation

4.1.1 Related Work

Researchers have developed different kinds of tactile sensors for over a decade. We refer the reader to two recent surveys of tactile sensing technologies for an overview [36, 38]. As has been noted in these survey articles, despite a long history of research in this area, tactile sensing systems on real robots have been less common, for example "hundreds of designs for tactile array sensors have appeared in the literature in the last 15 years, and a few of them are suitable for use with dextrous hands [36]" and "instead of inventing 'yet another touch sensor,' one should aim for the tactile sensing system [38]".

Examples of tactile sensors covering large surfaces of robot appendages on real robots have often used flexible printed circuit boards (PCBs) (e.g, [109, 37, 190]). The tactile sensor covering the forearm of our robot Cody, shown in fig. 8 on page 26 also uses flexible PCBs. These sensors are not easily adaptable to cover the joints of a robot because they do not stretch and can exhibit fatigue on repeated bending.

Prior research has also used resistive and conductive fabric, rubber and other flexible and in some cases stretchable materials to make tactile sensors (e.g, [150, 84, 16, 195, 80]). These are similar in spirit to our sensor, but most of these have not been demonstrated on a robotic system.

[16, 15] present a sensor that can be used to cover the elbow of a human or humanoid robot. The sensor uses electrical inverse tomography, which has the drawback of a complex inverse problem [211, 36]. [84] used conductive thread and fabric to make a whole-body tactile sensing suit. In that implementation, the sensor functioned as a binary contact detector.

In this chapter we present a straightforward design using multiple layers of conductive and resistive fabric. Our design is inspired by an open source project called rSkin [155].

4.1.2 Contact at the Joints Occurs Often in Software Simulation

For the experiments described in section 3.3.3 on page 33, within the software simulation testbed, we computed the percentage of trials in which the robot made contact with a cylindrical object at a joint at least once. Figure 24 illustrates the conditions when we count a contact as having occurred at the joint of the simulated robot. Table 5 on this page shows that with our controller for reaching in clutter, described in chapters 2 and 3 on pages 7–24, the simulated arm makes contact at the joints in approximately 15% to 47% of the trials depending on the total number of cylinders and the composition of the clutter.

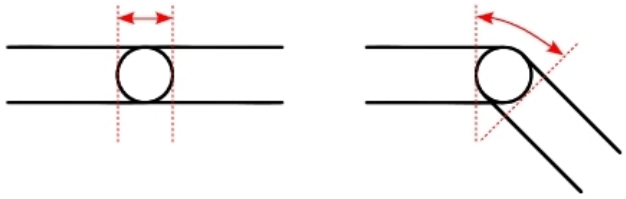


Figure 24: *For computing the results shown in table 5 on this page, if the arm made contact with an object at a location between the dashed red lines, we considered it a contact at a joint of the robot.*

Table 5: *Percentage of trials with contact at a joint in varying levels of clutter.*

	Total number of cylinders			
	40	80	120	160
25% movable, 75% fixed	36.7%	35.0%	20.7%	15.0%
50% movable, 50% fixed	37.2%	37.7%	30.0%	21.5%
75% movable, 25% fixed	41.7%	46.8%	38.5%	31.7%

4.1.3 Stretchable and Flexible Sensors to Cover Joints

Researchers have recognized the need for tactile sensors to stretch and bend to use them to cover the joints of a robot [190, 80]. In this section we present a simple kinematic model to motivate the requirement for tactile sensors that cover articulated joints to be able to bend and stretch.

We assume that the tactile sensor covers the outer surface of the arm and its end points are rigidly fixed to the two links at a distance of L_1 and $L2$ from the joint axis, as shown in fig. 25.

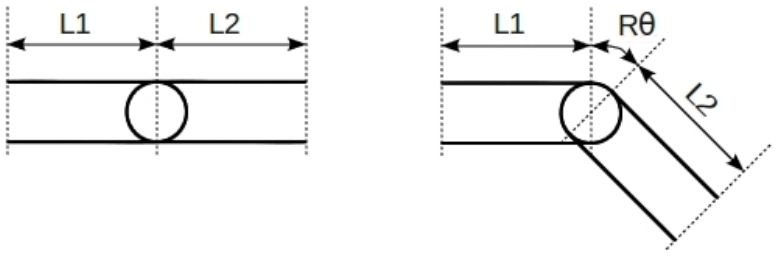


Figure 25: *Sketch of a single degree of freedom rotary joint to illustrate the requirements of bending and stretching to cover the joints of a robot with tactile sensors. As the joint rotates through an angle θ, the length to be covered with tactile sensors increases by $R\theta$ on one side of the arm, and decreases by $R\theta$ on the other. θ is in radians.*

From fig. 25, if the joint rotates through an angle θ, the length to be covered with tactile sensors increases by $R\theta$, where R is the radius of the arm at the joint. So, the ratio of the new length and the unstretched length of a tactile sensor covering the arm is given by

$$s = \frac{L_1 + L_2 + R\theta}{L_1 + L_2} \tag{25}$$

$$= 1 + \beta\theta, \tag{26}$$

where s is the amount by which the sensor would have to stretch and β is the ratio of the radius of the arm at the joint and the unstretched length of the tactile sensor ($\frac{R}{L_1+L_2}$). If we keep either θ or β constant, the stretch of the sensor is a linear function of the other variable.

Figure 26 on the next page shows the amount of stretch of the tactile sensor as a function of the joint angle for different values of β, the ratio of the radius of the arm at the joint and the unstretched length of the tactile sensor.

Other design possibilities to reduce or perhaps eliminate the requirement of sensor stretch for covering the joints include allowing the sensor to slide relative to the surface of the arm and fold similar to wrinkles in the skins of humans and animals. We have not investigated these options yet.

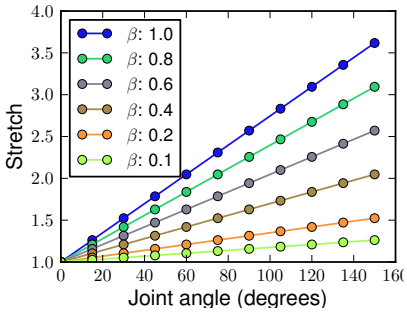


Figure 26: *The amount of stretch of the tactile sensor (s from eq.* (25) *on the previous page) as a function of the joint angle for different values of β, the ratio of the radius of the arm at the joint and the unstretched length of the tactile sensor.*

4.2 Sensor Design

4.2.1 Single Taxel

A single sensing element or taxel (short for tactile pixel) consists of five layers of fabric, illustrated in fig. 27. The layer in the middle is resistive fabric, which is sandwiched between two layers of conductive fabric. The resistance between the two electrodes made of conductive fabric decreases when we apply a force on the sensor.

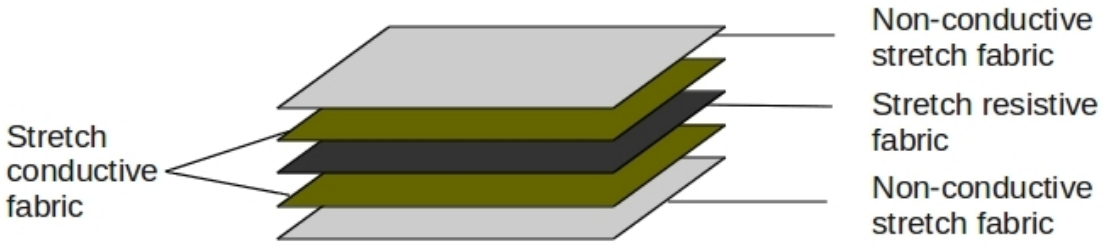


Figure 27: *Five layers of conductive, resistive and non-conductive fabric that make up our tactile sensor.*

The fabrics that we use for our sensor construction are as follows:

- ***Stretch conductive fabric:*** Silver plated 76% Nylon, 24% elastic fiber fabric purchased from LessEMF [11].
- ***Stretch resistive fabric:*** Knitted nylon/spandex fabric with proprietary conductive coating purchased from Eeonyx [10]. We use the LG-SL-PA fabric with a surface resistance of 16KΩ/sq.
- ***Non-conductive stretch fabric:*** Compression t-shirt for athletes. 80% nylon, 20% spandex and purchased from Amazon [9].

4.2.2 Signal Conditioning

In our implementation, we use each taxel as part of a resistive voltage divider as shown in fig. 28. We do not amplify the signal and directly use it as an input to a 10 bit analog to digital converter (ADC) by connecting it to one of the ADC input pins on an Arduino Mega 2560 R3 board [8]. We use R_v to denote the resistance of the taxel, with the subscript v referring to the fact that it is variable. For the second resistor of the voltage divider, we use the symbol R_1.

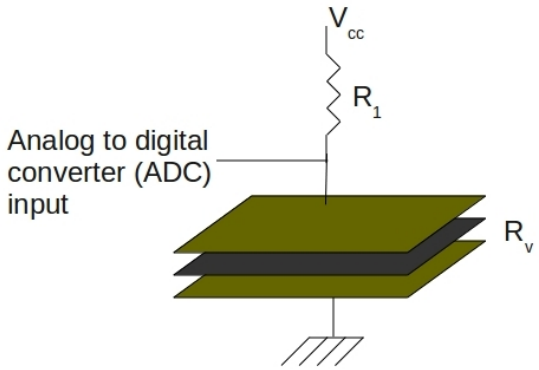


Figure 28: *In our implementation, use the taxel as part of a resistive of a voltage divider, and measuring the voltage across it with an analog to digital converter.*

In this chapter, we refer to the output of the analog to digital converter as the output of the tactile sensor. So, the output of the tactile sensor is given by $1023 \times \left(\frac{R_v}{R_v+R_1}\right)$, where 1023 comes from the fact that our analog to digital converter has 10 bits of resolution. Since R_v decreases when the force on the sensor increases, the output of the tactile sensor also decreases. Additionally, the resistive voltage divider has maximum sensitivity when R_1 equals R_v.

4.2.3 Tactile Sensor Array

To make an array of tactile sensors, we change one of the layers of conductive fabric to have multiple discrete electrodes of the desired shapes and sizes. The second conductive fabric serves as a common ground for all the taxels and we also use a single sheet of resistive fabric. Figure 33 on page 54 illustrates this for the tactile sensor array that we made for Cody.

4.3 Sensor Properties

In this section we investigate the relationship between the output of the tactile sensor and the total force or pressure applied to the sensor. We show that the tactile sensor output is not a function of either the total force or the pressure applied to the sensor. In fact, for a particular tactile sensor output, we can only place an upper bound on the total force and a lower bound on the pressure (depending on the area of the taxel). For the experiments in this section, we used $R_1 = 470\Omega$.

4.3.1 Test Setup

We made a square tactile sensor with sides of length 5cm and pushed down on it with a tool instrumented with an ATI Nano25 (with a calibration of SI-125-3) six-axis force-torque sensor. We varied the contact region to be a square with sides of length 1cm, 2cm, and 4cm by placing squares that we cut out of acrylic on the tactile sensor before pressing with the instrumented tool. To collect data, we pushed down on the taxels multiple times while keeping the tool vertical and logged the component of the force in the vertical direction measured by the force-torque sensor and the output of the tactile sensor. Figure 29 shows the different components of our test setup.

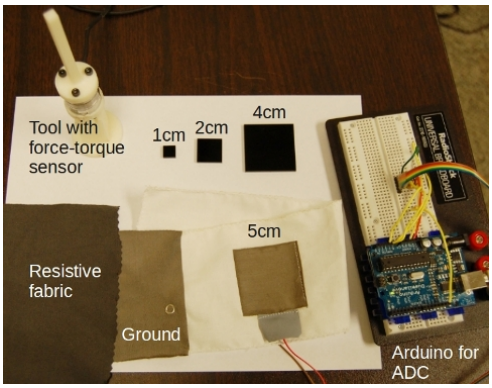


Figure 29: *Components of the test setup that we used to investigate some of the properties of the tactile sensor.*

4.3.2 Hysteresis

Figure 30 on the next page shows the tactile sensor output and the total force when we increased (loaded) and decreased (unloaded) the force on the tactile sensor two times. We see that the tactile sensor output does not show much noise but has different profiles for each loading and unloading cycle.

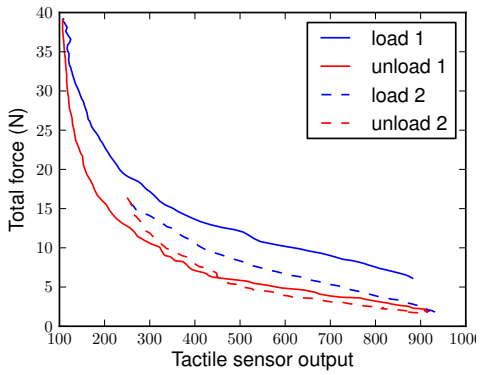


Figure 30: *Hysteresis in the tactile sensor output.*

4.3.3 Sensor Output and Total Force

Figure 31 shows the relationship between the total force applied to a tactile sensor (y-axis) and the output of the tactile sensor (x-axis) for different contact regions. For a particular output of the tactile sensor and contact region, we see a spread in the forces. We believe that hysteresis is a significant cause for this.

A more striking cause of the spread in the force for a given tactile sensor output is the variation in the contact region. A higher force distributed over a larger contact region produces the same tactile sensor output as a lower force over a smaller contact region. As a result, for taxels that have large areas, the uncertainty in the total force will be much greater. A smaller taxel would allow us to place tighter bounds on the maximum force for a given tactile sensor output.

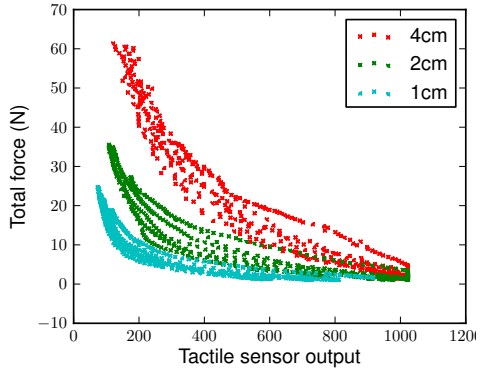


Figure 31: *Relationship between total force and tactile sensor output. The red, green and cyan scatter plots correspond to square contact regions with sides of length 4cm, 2cm and 1cm respectively.*

4.3.4 Sensor Output and Pressure

Figure 32 shows the relationship between the pressure (y-axis) and the tactile sensor output (x-axis). Similar to the relationship between the total force and sensor output, we see a spread for a particular contact region, and a much greater spread across contact regions.

A higher pressure over a smaller contact region produces the same tactile sensor output as a lower pressure over a larger contact region. To generate this plot we assume that the force is uniformly distributed over the contact region and compute the pressure as the total force divided by the contact area.

This property is not unique to our sensor design. As described in [190], the output of a capacitive tactile sensor can be identical for different pressures if the probes have different contact area. In our current design, because the spatial resolution is low, this effect can be more significant.

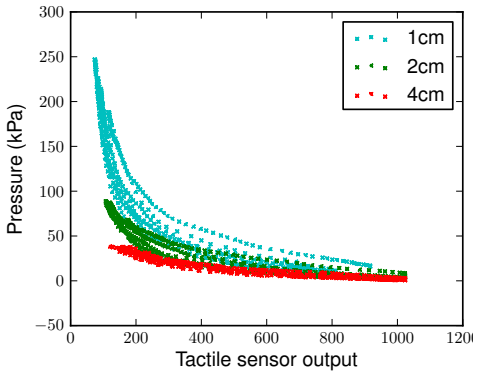


Figure 32: *Relationship between pressure and tactile sensor output. The red, green and cyan scatter plots correspond to square contact regions with sides of length 4cm, 2cm and 1cm respectively.*

4.3.5 Tactile Sensors in Practice

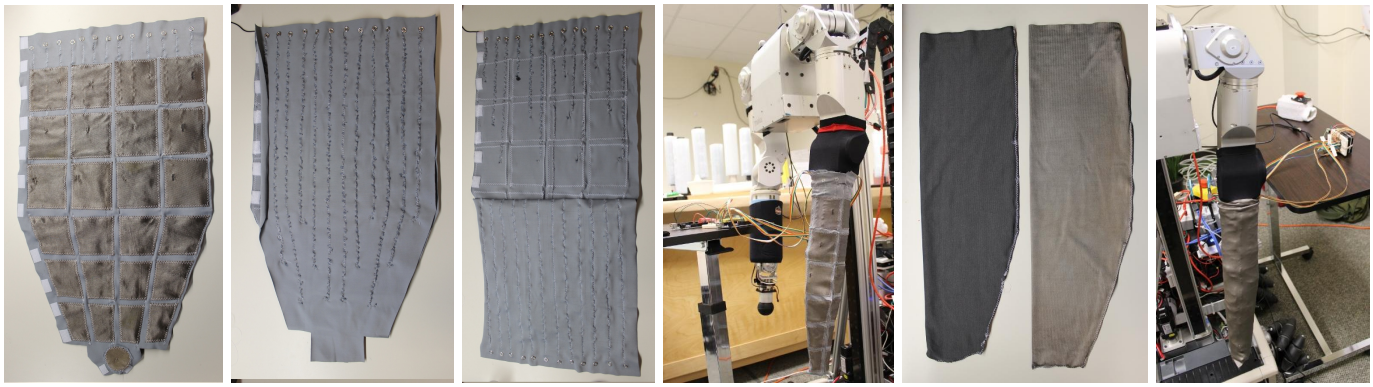
Despite the fact that the tactile sensors have hysteresis and the output is a function of neither the force nor the pressure, we have found them to be promising. The sensors are inexpensive and simple to fabricate. We could rapidly prototype different sensor arrays and cover large regions of two robot arms as we describe in the rest of this chapter. The sensors respond well to changes in the force and in our initial tests, we have found that our controller can use this tactile feedback to perform useful tasks while keeping the forces low. Additionally, since it is made of layers of fabric, the sensor does not get easily damaged if it is bent, folded, and stretched.

4.4 Tactile Sensing Sleeve for Cody

4.4.1 Sleeve Design

We designed the tactile sensor array as a single sleeve with 25 taxels that covers the end effector, wrist joints, and forearm. Our goal with this design was to prototype a sensor that covers the wrist joints of the robot. For Cody, the last two wrist joints have relatively small ranges of motion, $\pm 45°$ for wrist flexion/extension and abduction/adduction. This, along with the small radius at the wrist joint (relative to the overall length of the sensor), suggests that the requirements for sensor stretch will not be very severe according to the model of section 4.1.3 on page 48.

Figure 33 shows the sleeve that we built. We used conductive thread as stretchable wire to connect the conductive fabric electrodes to female snap buttons on one end of the sleeve. We then soldered normal wires to male snap buttons and connected the wires to the resistive voltage divider. The conductive thread itself does not stretch. However, we sewed it in a zig-zag pattern, also known as a stretch stitch, which lets the fabric stretch although we have sewn conductive thread through it. Figure 33b shows this pattern.

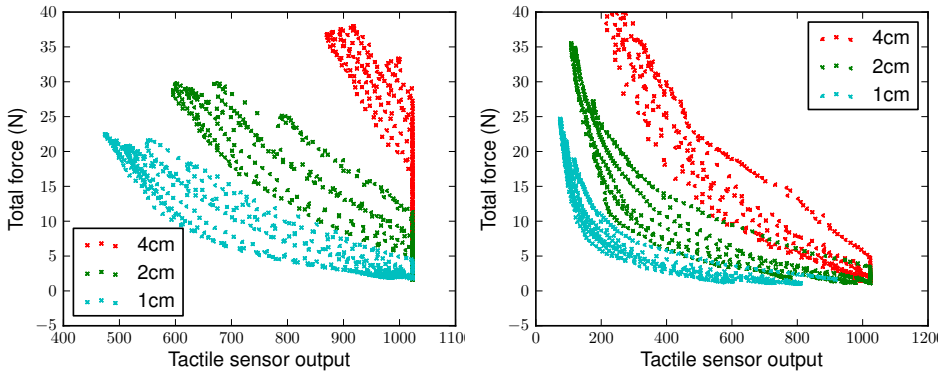


(a) (b) (c) (d) (e) (f)

Figure 33: *Different layers of the tactile sensing sleeve that we made to cover the forearm, wrist and end effector of Cody.* ***(a)*** *Layer with 25 electrodes of conductive fabric.* ***(b-c)*** *The other side of this layer shows conductive thread to connect each electrode to a row of snap button connectors at one edge.* ***(d)*** *Layer with 25 electrodes mounted on the robot.* ***(e)*** *Two additional sleeves of the resistive fabric (left) and conductive fabric (right).* ***(f)*** *Robot arm after we have put the resistive and conductive fabric sleeves.*

4.4.2 Sensor Output when the Wrist Joints Move

Although the fabrics that make up our sensor are flexible and stretchable, there are internal stresses when the wrist joints move. We have empirically found that our choice of R_1 of fig. 28 on page 50 affects how much these internal stresses appear in the tactile sensor output. For example, with R_1 = 47Ω, we do not see any significant change in the tactile sensor output due to the motion of the joints. However, with R_1 = 470Ω, the tactile sensor output changed significantly when the joints moved.



(a) R_1 = 47Ω (b) R_1 = 470Ω

Figure 34: *The tactile sensor output varies with different values for R_1.* ***Left:*** *For small values of R_1, the sensor does not respond to low to moderate forces that are distributed over a large area.* ***Right:*** *For higher values of R_1, the sensor is sensitive to low forces over small areas but less sensitive to moderate forces over a small area.* ***Note:*** *The y-axes are identical for the two plots but the limits on the x-axes are different.*

To avoid noise in the sensor output due to motion of the joints, we selected a value of 47Ω for R_1. This has the disadvantage that the tactile sensor becomes insensitive to low to moderate forces over a large area, as shown in fig. 34a. Another approach to reducing noise in the sensor output from joint motion could be to use predictions of the tactile sensor output using the state of the robot to filter the signal. We have not investigated this approach in this work.

4.4.3 Testing the Sensor – Reaching in a Pipe

As a demonstration of the tactile sensing sleeve, we set up a trial where the robot had to reach to a goal location that was inside a pipe. Figure 35 on the following page shows the robot Cody, and the visualization of the tactile sensors after the robot reached the goal location. Appendix A.2 on page 147 provides a link to the video associated with this trial.

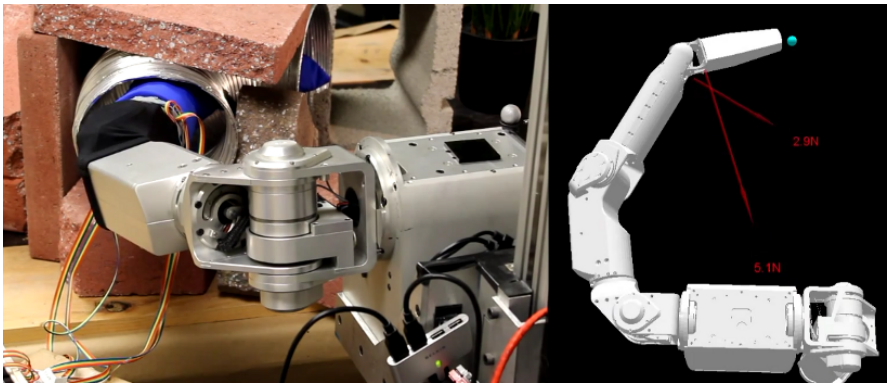


Figure 35: ***Left:*** *Cody reaching to a goal location inside a pipe.* ***Right:*** *Visualization of the output of the fabric tactile sensor array*

4.5 Tactile Sensors for the PR2

For the PR2, we decided to focus on covering large surfaces of the arm, including the gripper, forearm and upper arm with tactile sensors. We did not attempt to cover any of the joints with sensors for our current prototype. We made this decision for two reasons. First, the range of motion for the elbow flex and the radius of the elbow joint on the PR2 is much larger than the wrist joints on Cody. A sensor covering the elbow of the PR2 would have to stretch by a very large amount.

Second, the wrist and forearm roll joints would result in torsional stress on the fabric sensor. If the sensor were a single sleeve like our design for Cody, these joints would stretch the sensor very severely. A single sleeve design may not work and we might have to attempt to mount modular sensors in such a way to avoid this torsional stress. We decided to circumvent these potential problems for our current sensor prototype.

Our whole-arm sensor design consists of modular sub-components that we mount to the upper arm, forearm, and left and right link of the gripper. We have 3 taxels on the upper arm, 14 taxels on the forearm, and 16 taxels on the gripper (10 of which are our custom fabric sensors and the remaining 6 of use the existing finger tip pressure sensor array on the PR2). Figures 36 to 38 on pages 57–58 show the whole-arm tactile sensor on the PR2. We used a value of 470Ω for R_1.

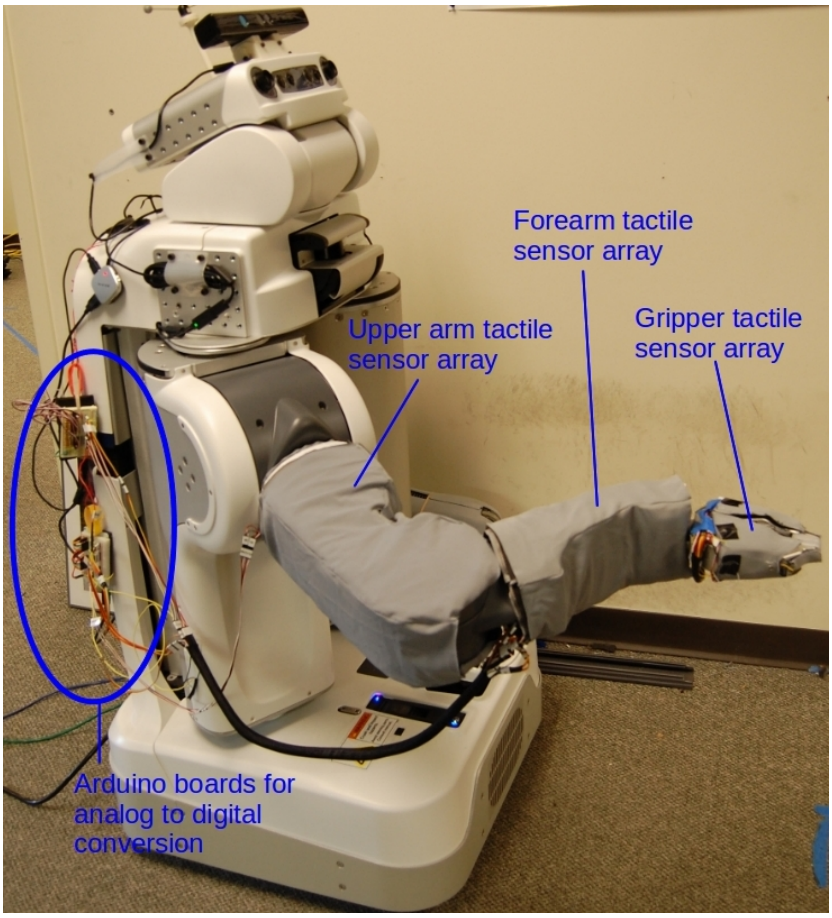


Figure 36: *Current implementation of tactile sensor arrays for the upper arm, forearm, and gripper of the PR2.*

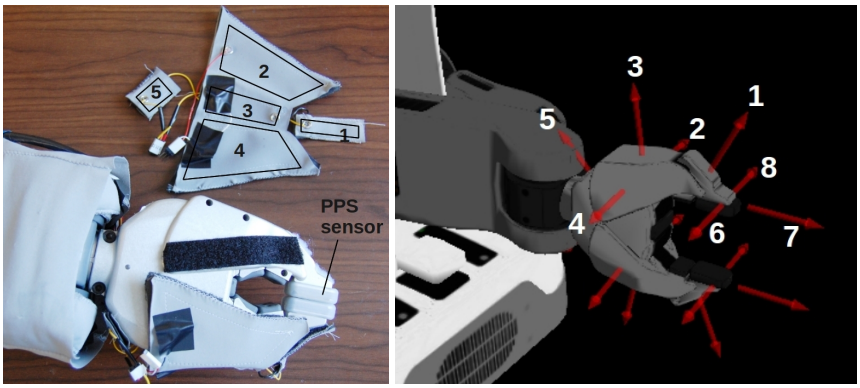


Figure 37: ***Left:*** *We have fabricated the gripper tactile sensor array in two parts with 5 taxels each. We attached these to the gripper using Velcro fasteners. The individual taxels are numbered and the black outline show the spatial extent of the taxels.* ***Right:*** *We report contact anywhere on a taxel as contact at the locations of the red arrows. The white numbers match the taxel numbers on the left image. The arrows labeled 6-8 correspond to pressure sensing fingertips by Pressure Profile Systems (PPS) that are part of the sensor suite on the Willow Garage PR2.*

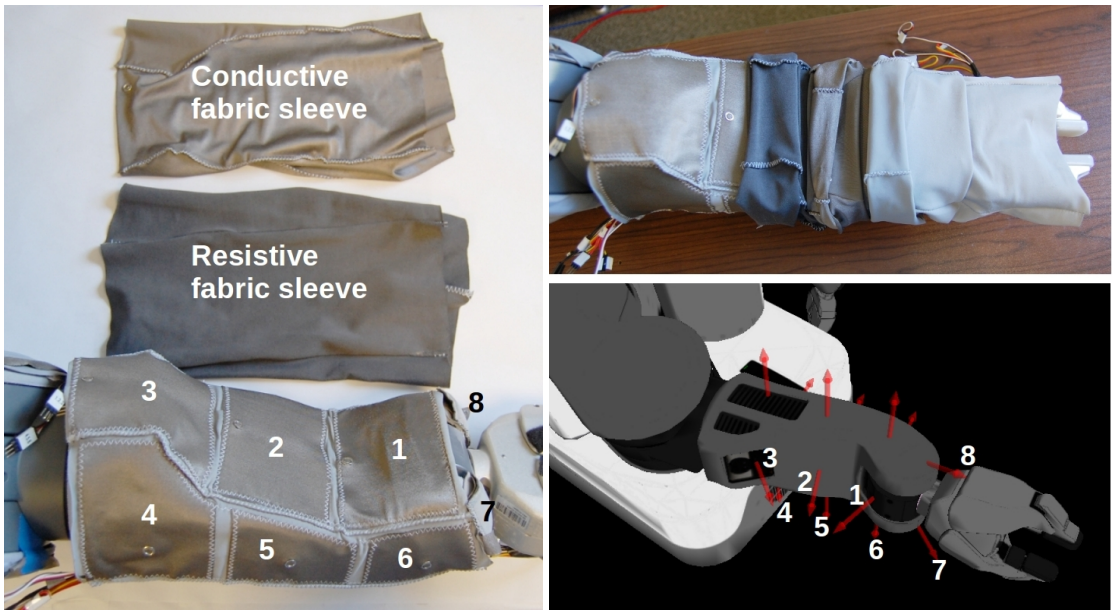


Figure 38: ***Left:*** *We have three taxels along the length of each of the four surfaces of the forearm. For example, the taxels numbered 1-3 are on the top surface. Additionally, we have two taxels on the front surface of the forearm (7 and 8).* ***Right Top:*** *The different layers of fabric that make up the forearm tactile sensor array.* ***Right Bottom:*** *Visualization of the contact locations and direction of contact force that our sensor reports for each taxel.*

4.6 Preliminary Results with Teleoperation

In this section we present some preliminary results with teleoperating a PR2 with whole-arm tactile sensing that illustrate some future opportunities for robots with whole-arm tactile sensing and controllers that explicitly allow contact with the world.

4.6.1 The System

Figure 39 on the following page shows the interface used by the remote operator to command the robot. The operator sees a 3D model of the robot, and can optionally take a snapshot of a point cloud from a Kinect sensor mounted on the robot. The operator can rotate, pan, and zoom the 3D model of the robot and the point cloud. This functionality uses `rviz` and `pr2_interactive_manipulation` that are part of the Robot Operating System [170].

The remote operator can also move and orient a rendered gripper by clicking and dragging on the arrows and circles around it. A right click on the arrows displays a menu with these options:

- **Go:** Sets the goal location $\in \Re^3$ for the finger tips to be the finger tips of the rendered gripper. The robot uses the model predictive controller described in chapter 2 with feedback from the whole-arm fabric tactile sensors and attempts to reach the goal location while keeping the

contact forces low. The controller ignores the orientation of the rendered gripper.

- **Orient:** Sets a goal location and orientation for the model predictive controller. The extension to the controller to allow a goal orientation is work by Marc Killpack and is not described in this thesis.
- **Stop:** Stops the model predictive controller so that it does not update the virtual trajectory of the "simple" impedance controller, described in section 2.3.2.1 on page 15.
- **Open Gripper:** Opens the gripper.
- **Close Gripper:** Closes the gripper.
- **Zero Skin:** Uses the current sensor values as the bias. This allows the operator to zero out small phantom sensor values that can show up due to relative motion of the different layers of fabric, cables that are currently routed on the outside pushing against a taxel, hysteresis, and signal drift.

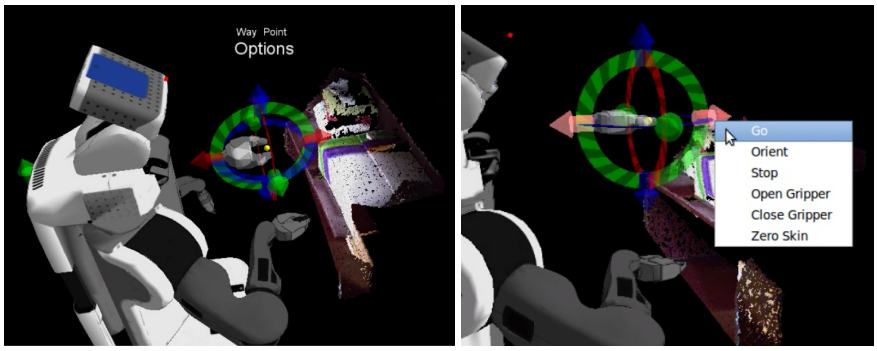


Figure 39: *Teleoperation interface that the remote operator uses.*

4.6.2 Example 1 – Manipulate on a Shelf

For this trial, we used the system described in the previous section to teleoperate a PR2 to grasp a pill bottle from a shelf and put it in a basket on a lower shelf. As the remote operator, we only used the visualization within the teleoperation interface. As illustrated in fig. 40 on the following page, our model predictive controller with feedback from the whole-arm tactile sensors enabled the remote operator to complete the task with relatively high level commands. Appendix A.2 on page 147 provides a link to the video associated with this trial.

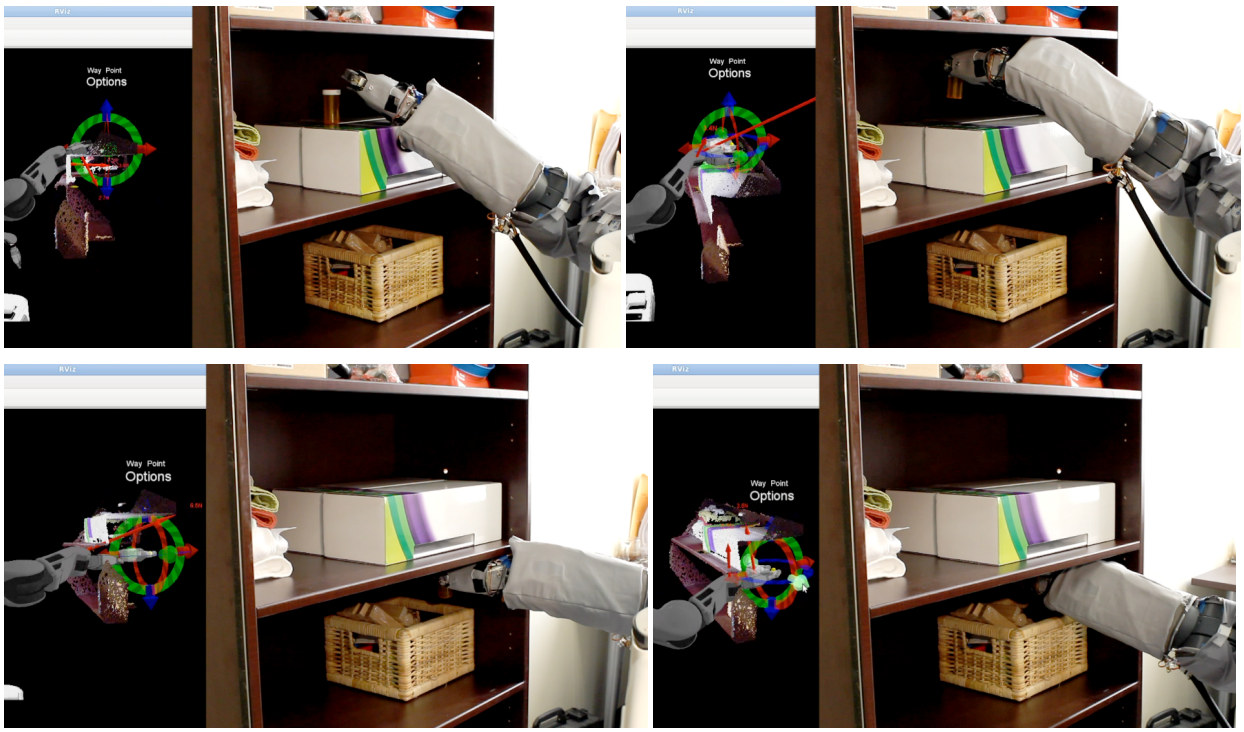


Figure 40: *Images showing the PR2 making contact with the shelf at the forearm and gripper while we commanded it to reach different goals on a shelf. The rendering on the left of each image is what the remote operator sees. The red arrows on the 3D model of the arm indicate tactile sensor output.*

4.6.3 Example 2 – Manipulate Around a Person's Body

This example has been motivated in part from the Robots for Humanity project, which is a collaboration between the Georgia Tech Healthcare Robotics Lab, Willow Garage, Prof. Bill Smart, and Henry Evans. As part of this project, Kelsey Hawkins, Philip Grice and Prof. Charles C. Kemp are developing a system to enable Henry, who has quadriplegia, to teleoperate a PR2 close to his face and body and perform tasks such as shaving and scratching. This system currently uses a wrist-mounted force-torque sensor and restricts contact to the end effector of the robot.

On June 26, 2012, Henry used the teleoperation system with whole-arm tactile sensing described in section 4.6.1 on page 58 and moved PR2 in close proximity to and in contact with his body, as shown in fig. 41 on the next page. During these initial tests, Henry had the following feedback:

- "I think its a good safety feature because it hardly presses against me even when I tell it to"
- "What would be interesting is to shave with skin"
- "It really feels safe to be close to robot"
- "I like it"

The following day Henry attempted two tasks with the same system. First, he attempted to grasp a hand towel and bring it close to his face to wipe his mouth. Next, he tried to pull up a blanket. With a couple of retries, Henry was able to accomplish both these tasks which involved the robot making contact with his body and the environment. Figures 42 and 43 show some images from these trials. Appendix A.2 on page 147 provides a link to videos associated with these trials.

Figure 41: *Henry Evans teleoperating a PR2 while it is in contact with his body.*

Figure 42: *Henry Evans teleoperating a PR2 to grasp a hand towel and bring it close to his mouth.*

Figure 43: *Henry Evans teleoperating a PR2 to pull up a blanket.*

A week after the tests, Henry provided the following additional feedback about our system:

1. “Skin

    - Overall awesome
    - Feels VERY safe
    - Faster than motion planning
    - It just wriggles around obstacles”

2. “DEFINITELY keep developing this !”

4.7 Conclusion

In this chapter, we presented the design and two initial prototypes of tactile sensor arrays made of multiple layers of stretchable conductive, resistive and non-conductive fabric that we used to cover large surfaces of two robot arms. Despite the large number of tactile sensor designs in the literature and some commercially available sensors, we believe that our sensor is promising. The advantages of our sensor design relative to others are lower cost, ease of fabrication and signal conditioning, flexibility and stretchability of the sensor, and the potential to rapidly try out multiple designs to cover large surfaces of robot arms. The limitations of our current implementation include uncompensated hysteresis effects, low spatial resolution, and reduced sensitivity when covering the joints of a robot.

With an informal experiment, we showed that Henry Evans, who has quadriplegia, was able to use our controller of chapter 2 and this tactile sensor to teleoperate a PR2 robot in close proximity to, and in contact with his body. We also presented a single trial of manipulating an object on a shelf with multiple contacts between the robot arm and the environment.

We plan to release our design and code as open source (appendix A.2 on page 147) with the hope that other roboticists replicate and improve it and add whole-arm tactile sensing to their robots.

CHAPTER V

DOOR OPENING

A large variety of doors and drawers can be found within human environments. Operating these mechanisms plays a role in many daily activities, such as moving within an environment or retrieving an object that has been stored. Being able to operate these same mechanisms would help service robots assist with similar activities.

Within this chapter we present a set of feedback controllers that enable a mobile manipulator to reliably and autonomously approach and open doors and drawers for which only the location and orientation of the handle have been provided. The robot uses low stiffness compliant actuation at the joints, a wrist-mounted force-torque sensor, and a form of joint space impedance control referred to as "simple" impedance control [77] or equilibrium point control [92].

Our controllers compute online virtual trajectories for the "simple" impedance controller to twist door handles and push open doors, and hook on to handles and pull open doors and drawers. The trajectory is called virtual because there is no requirement for the actual trajectory of the arm to match the trajectory of the impedance controller. The arm simply behaves as if it is connected to the virtual trajectory via low stiffness torsional visco-elastic springs at the joints.

Our robot pushed open 8 different doors in 32 trials with an 87.5% sucess rate, pulled open 10 different doors and drawers in 40 trials with a 92.5% success rate while estimating the mechanism kinematics online. We empirically demonstrate that our controllers are robust with respect to variations in the mechanism, the pose of the base, the stiffness of the arm, the friction between the door handle and the end effector, and the way the handle was hooked. We also demonstrate that our controllers can coordinate the movement of the robot's omnidirectional base and compliant arm while pulling open a door or drawer.

The work presented in this chapter has previously appeared in the proceedings of the *IEEE-RAS International Conference on Humanoid Robots* [92], *IEEE-RAS International Conference on Robotics and Automation* [94], and the *RSS Workshop on Mobile Manipulation in Human Environments* [91].

5.1 Related Work

Researchers have been developing systems for autonomous manipulation of doors for more than a decade [145, 147, 156]. Within the task of opening doors, researchers have built systems to open doors between rooms, as well as pull open cabinet doors. Many of these systems either require a map [158, 175], training [114, 169], or an explicit door model [189, 160]. Additionally, much of the previous work on door opening has focussed on only one or two aspects of the complete task [28, 117, 55, 164]. Instead, the systems presented in this chapter integrate perception, navigation and manipulation to move up to the door handle and open the door.

Contemporaneous research demonstrated performance similar to our system [114, 148]. Researchers have developed a system that can pull open doors between rooms [34], and one that can navigate an office environment by pushing open doors after twisting the door handle [215]. Other research has demonstrated a specialized gripper for a wheelchair mounted robot arm that is capable of twisting a variety of door handles and knobs [173].

Researchers have also developed systems to open cabinet doors and drawers. Some of these require an explicit door model [44, 189]. Others [221, 165, 163, 128] are similar in spirit our feedback controllers, but are slow, or lack empirical testing on a number of different mechanisms.

The contribution of this thesis to the literature on door opening is that we have empirically demonstrated a system that can autonomously pull open a number of different cabinet doors and drawers, robustly and efficiently, without prior knowledge of the mechanism kinematics and forces, and can use the mobile base to increase the workspace of the robot. Further, as described in chapters 6 and 7 on pages 99–116, we have created a database of forces applied to the handles while pulling open doors and drawers. We show that this database can be used to inform tradeoffs in assistive robot design, as well as enable a robot to build data-driven models of the forces that it should expect, and better detect anomalous conditions.

5.2 The Robot, Cody

For the work described in this section, we use the robot Cody. We described the main components of the robot in section 1.2 on page 3.

For this work, the robot uses two distinct types of sensors. First, the robot uses a laser range finder (Hokuyo UTM-30LX) mounted on a servo motor (Robotis Dynamixel RX-28) on top of the torso. The servo tilts the laser range finder about the horizontal axis (fig. 44). We use this tilting laser range finder to obtain 3D point clouds of the environment. The laser range finder has a resolution

of 0.25°and we obtain planar scans at 20Hz. The servo encoders have a resolution of 0.3°. This sensing configuration was inspired by the Willow Garage PR2. We used this sensor before there were commercially available sensors such as the Microsoft Kinect. We provide links to the code and hardware designs for this tilting hokuyo sensor in appendix A.3 on page 147.

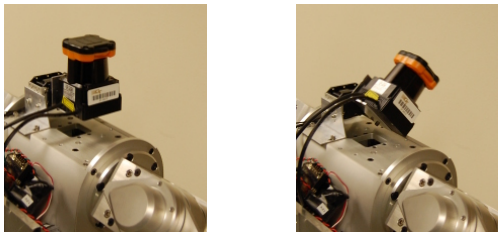


Figure 44: *A Hokuyo UTM-30LX mounted on a Robotis servo that enables the robot to capture 3D point clouds.*

Second, the robot senses forces and torques using a wrist-mounted 6-axis force/torque sensor (ATI Mini40 from ATI Industrial Automation). The arms joints also sense torque, but the current behaviors only use this sensing implicitly in the context of "simple" impedance control.

5.3 Control Structure

All the controllers for the compliant arm use a form of joint space impedance control that is referred to as "simple" impedance control or equilibrium point control, as described in section 2.3.2.1 on page 15. In contrast to chapter 2, the controllers described in this chapter do not use model predictive control. Instead, they either use linear Cartesian-space virtual trajectories or a hand coded control law that uses knowledge of the constrained kinematics of the task.

Figure 45 on the next page shows the common structure for all the feedback controllers that we present in this chapter. At every time step, the controllers compute an incremental update to the Cartesian-space virtual trajectory, $\Delta x^v \in \Re^3$. The controller accumulates this change over time to generate a Cartesian-space virtual trajectory, x^v. This denotes the position that the end effector would settle to if the arm were moving in free space and the gravity compensating torques were perfect. We then use the inverse kinematics solver from KDL [199] to compute a joint-space virtual trajectory, ϕ, that is input to the "simple" impedance controller. For all the controllers in this chapter, we use a fixed orientation of the end effector that would point the end effector away from the torso and normal to its from surface for the inverse kinematics computation.

We present two distinct systems for opening doors and drawers. First, in section 5.4 on the following page we present a system that can detect door handles with a 3D scan from an actuated laser range finder, servo up to the handles, twist them and push open doors. In section 5.6 on page 78

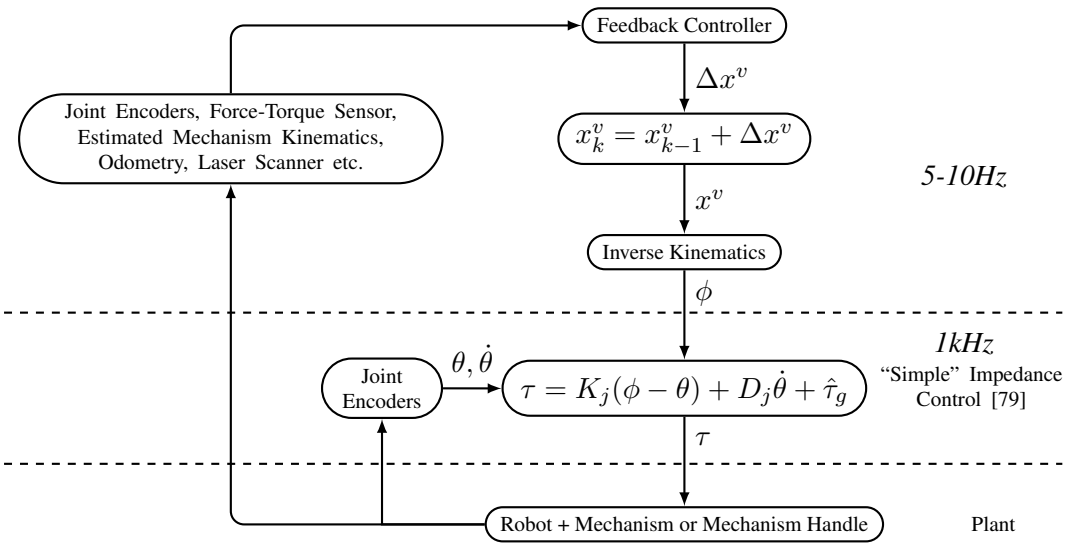


Figure 45: *Block diagram of the common control structure for the different feedback controllers that we present in this chapter. The controller frequencies are specific to our implementation.*

we present feedback controllers to pull open doors and drawers. In section 5.7 on page 85 we extend the controllers of section 5.6 to enable the robot to move an omnidirectional base while opening a mechanism, thereby increasing its effective workspace and improving the performance.

In the rest of this chapter, we describe these two systems, including the components for perception and navigation, and then present our evaluation of the individual components and the integrated systems.

5.4 Twisting Door Handles and Pushing Open Doors

In this section we describe an integreted system that can navigate up to a door, twist the door handle, and push open the door through a small angle.

5.4.1 System Overview

Figure 46 on the next page shows the different steps that the robot goes through to navigate to a door handle and push open the door. The input to the door opening behaviors presented in this paper is a rough 3D location for the door handle in the robot's coordinate frame. We have previously presented interfaces (laser pointer interface and a touch screen) with which users can select a 3D location in the world [35]. More generally, this location could be provided by an autonomous perception system. In this work, we display a scan from the laser range finder and allow the user to click on any point to designate the location of the handle. We fix the initial estimate of the height of the door handle to 1m.

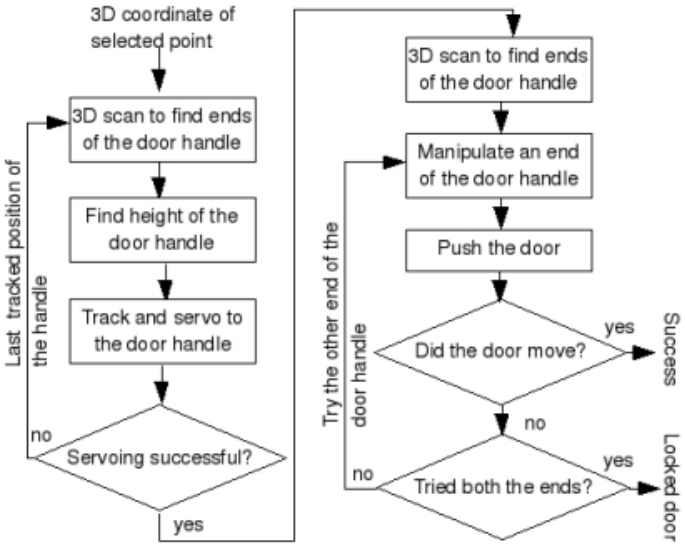


Figure 46: *The different behaviors that the robot executes to open a door.*

In our experiments, the robot starts around 1.5m from the door handle and the user selects a 3D location in the neighborhood of the door handle. The robot then makes a 3D scan with its tilting laser range finder (fig. 44 on page 65), segments the door handle (section 5.4.2.3 on page 69) and lowers the laser range finder so that it scans at the height of the door handle (section 5.4.2.4 on page 69). The robot then servos to the door handle (section 5.4.3 on page 70) and takes another 3D scan. After segmenting the door handle, the robot haptically finds the surface of the door and makes contact with the door handle near the appropriate tip (section 5.4.4.1 on page 71). Finally, the robot twists the door handle and pushes the door open (sections 5.4.4.2 and 5.4.4.3 on pages 71–72).

5.4.2 Perceiving the Door

The robot uses its laser range finder to make estimates about different task-relevant features related to the door. The robot's perception starts with raw sensor data (a point cloud consisting of $\sim$ 40,000 3D points or a planar laser scan of around 600 points) and reduces it to low-dimensional, task-relevant features: the estimated location of the door handle, the two tips of the handle (left or right), the estimated height of the handle above the ground, and the estimated orientation of the door. Our methods for segmenting and detecting the door and door handle have similarities with [182] and contemporaneous work on perceiving doors [181], although we only perform these perceptual tasks within a small volume around a user-selected 3D location.

5.4.2.1 Estimating Door Orientation and Distance

The robot first estimates the orientation of the door (with respect to an axis of rotation parallel to gravity) and the distance of the door from the robot using a non-tilting scan from the laser range finder. In addition to this scan, the algorithm requires that a location near the door be provided. The orientation and distance estimates are used when approaching the door and when detecting and segmenting the handle.

The algorithm assumes that the door is wider than 0.3m, so it first finds line segments of length greater than 0.3m in the laser scan using a Hough transform from OpenCV (see fig. 49 on page 70).

For each line segment, it then finds the shortest distance between the provided location and the line segment. The algorithm rejects segments with a distance greater than 0.5m. It also assumes that the robot is within 45° of facing the door and throws out any line segments that violate this assumption. Finally, it assumes that the door will either be recessed or flush with the walls, so the algorithm returns the distance and orientation of the line segment whose perpendicular distance from the robot is the greatest.

5.4.2.2 Detecting the Door

The door detection algorithm takes as input a 3D point cloud and a location that is believed to be less than 20cm from the door handle. This algorithm and the handle segmentation algorithm that follows use a coordinate frame whose origin is at the robot with the floor parallel to the X-Y plane and the door parallel to the Y-Z plane, as shown in fig. 47 on the following page. The robot maintains this coordinate frame by estimating the orientation of the door in the X-Y plane as described in section 5.4.2.1. Both algorithms only analyze points from the 3D point cloud that fall within a volume of interest (VOI). This VOI is an axis aligned box of 40cm x 60cm x 40cm along the X, Y, and Z axes and is centered on the provided location near the handle.

The detection algorithm assumes that the door is planar, parallel to gravity within the VOI, and is represented by a significant number of points in the VOI. Since the robot already has an estimate for the door's orientation in the X-Y plane, these assumptions imply that it only needs to estimate the distance of the door from its origin along the X-axis. To do this, the algorithm uses a histogram in which each bin covers a mutually exclusive 5mm range of depths along the X-axis in the VOI. It then estimates the depth of the door as the depth corresponding to the bin with the largest number of entries.

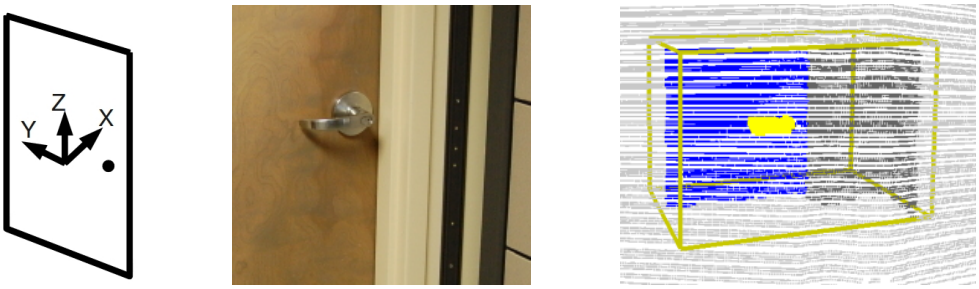


Figure 47: ***Left:*** *Coordinate axes parallel to the axes of the frame with the origin at the robot that is used by the door and door handle detection algorithms.* ***Middle:*** *Image of a door and handle.* ***Right:*** *The output of the door handle segmentation algorithm. The volume of interest (VOI) is the yellow box, the door is blue, the door handle is yellow, and the points inside the VOI are dark grey. The point cloud outside the VOI is light grey.*

5.4.2.3 Segmenting the Door Handle

Once the robot has found the plane associated with the door, it finds the handle protruding from its surface. To segment the door handle within the VOI, the handle segmentation algorithm removes all points on and behind the plane of the door, and any point less than 3.5cm in front of the detected plane of the door. It then converts the remaining point cloud into a 3D occupancy grid. The resolution of the grid is 2cm along the X and Z axes and 0.25cm along the Y-axis. The algorithm clusters these grid cells into objects by performing a 3D connected components labeling with 26-connectivity.

The algorithm assumes that the door handle will not be shorter than 4cm (dimension of the cluster along the Y-axis) and will not be wider than 10cm (dimension of the cluster along the Z-axis), so it throws out all clusters whose dimensions violate these assumptions. It then selects the cluster closest to the robot's initial estimate of the door handle location as the door handle. Figure 47 shows the output of the door and door handle segmentation algorithms.

5.4.2.4 Estimating the Door Handle's Height

In this implementation, the laser range finder needs to scan parallel to the ground at the height of the door handle when servoing to the door handle. To estimate the height of the handle with high precision, the robot scans parallel to the ground while the linear actuator moves the robot's torso down. Points on the scan line that are to the right or left of the segmented handle are removed from the scan. While descending, the robot stores the average distance of the remaining points from the estimated surface of the door. The height with the maximum average distance is estimated to be the door handle's height.

5.4.3 Approaching the Door

The robot first performs a 3D scan of the area surrounding the user-selected location and segments the door handle. This segmentation gives the robot the approximate position of the door handle and the positions of the tips of the door handle in its frame of reference. With reliable odometry, this would likely be sufficient for the robot to approach the handle. For this system, we do not have reliable odometry estimates due to slipping by the Mecanum wheels. Consequently, we servo the robot to the handle using the laser range finder.

Figure 48: *These images show the robot servoing to the door handle.*

Figure 49: ***Left:*** *Position of the robot (red circle), the points from the laser scan (dark blue), the estimate of the door (green line) and the tracked position of the door handle (black circle) during the servoing behavior.* ***Right:*** *(clockwise from the left) Laser scan rotated and converted into an image, the template of the door handle and the result of correlation of the template with the image.*

Prior to servoing, the robot lowers its laser range finder to look directly at the handle using the method described in section 5.4.2.4. It then uses its omnidirectional base to simultaneously and independently servo its orientation and its position based on the estimated orientation of the door and the estimated position of the handle. The robot continues to servo to the door handle until it is within the workspace of the door opening controller for the arm (see fig. 48).

To estimate the position of the handle while servoing, the robot tracks the handle. The handle tracker converts the laser scan into an image and uses a template and 2D image correlation to estimate the handle's location. Prior to this conversion the tracker rotates the laser scan into a canonical orientation based on the orientation estimate for the door. The tracker acquires its initial

template from the estimated location of the door handle tips from the handle segmentation (see fig. 49). As it tracks the handle, it updates the template. If the tracker detects that it has lost the handle, due to a jump in the handle's estimated pose (change exceeds 0.1m), the robot rescans the environment and tries again. For our experiments the robot is only allowed to try again once.

5.4.4 Manipulating the Door

Once the servoing has stopped, the robot assumes that it is facing the door, and the door handle is within the workspace of the door opening controller for its right arm.

All the controllers in this section update the Cartesian-space virtual trajectories as described in section 5.3 on page 65. Additionally, these controllers use feedback from the wrist force-torque and the laser scanner to determine stopping conditions and success or failure.

5.4.4.1 Making Contact with the Handle

The robot reaches out towards a position on the door that is above the desired contact location on the handle. It first uses the linear actuator to raise the arm up towards this location. It then positions the arm laterally so that it can reach out using a trajectory normal to the door that passes through this target location. Finally, it reaches out toward the position and stops when it detects contact with the door. It then moves the end effector away from the door by 2cm and uses the linear actuator to move down until the end effector makes contact with the handle near the tip. During this downward movement, the arm is held in a fixed compliant posture. This is shown in fig. 50.

Figure 50: ***Left:*** *The robot reaches out to haptically detect the surface of the door and then moves down until it makes contact with the handle near the tip.* ***Middle:*** *The robot twists the handle.* ***Right:*** *The robot pushes open the door.*

5.4.4.2 Twisting the Handle

Figure 50 shows the robot twisting a door handle in the counter-clockwise direction. The twisting controller sets Δx^v as a vector of constant length (0.5cm) oriented along a line in the plane parallel to the surface of the door. For most of our experiments, the line has an angle of 20° or -20° with the vertical, depending on which tip of the handle the robot is manipulating (left or right).

To decide which tip of the door handle to manipulate first, we use a heuristic based on the estimated line segment of the door (section 5.4.2.1 on page 68). If the handle is closer to the right end of the line segment, the robot manipulates the left end of the handle and vice-versa. For a flush door, the line segment may not be informative. To compensate for errors in estimating the extent of the door, the robot tries both the ends of the door handle.

In addition to the line parallel to the plane of the door, the twisting behavior also changes the Cartesian-space virtual trajectory in the direction normal to the door with the goal of maintaining a pushing force of 10N. It uses a bang-bang controller that at each time step adds a vector of length 0.5cm to Δx^v which points out of the door if the pushing force exceeds 10N, and into the door if the force falls below 10N.

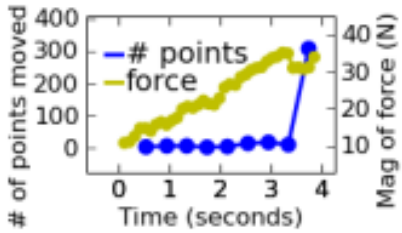


Figure 51: *Motion of the door as observed by looking for points that moved in the laser scans (blue) and the magnitude of the contact force at the end effector (yellow) while the robot twists the door handle. The X-axis is the time in seconds (twisting the door handle took around 4 seconds).*

While it twists the door handle, the robot monitors its progress using the laser range finder, which scans parallel to the ground, and the wrist-mounted force/torque sensor. The robot looks for motion of the door by comparing the current laser scan with a scan taken just prior to twisting. It subtracts these scans and counts the number of points that have moved by more than 0.5cm. The pushing force of 10N that the twisting controller tries to maintain results in a sharp increase in this count when the handle is twisted enough, as shown in fig. 51. The twisting controller continues until either the robot detects significant door motion, the magnitude of the total resultant force exceeds 45N (indicating that the robot has twisted the handle to the end stop or that the door is locked), or the magnitude of the force falls below 2N (indicating that the end effector has slipped off the handle).

5.4.4.3 Pushing the Door Open

Without moving the mobile base, the robot attempts to push the door open with a straight line Cartesian-space virtual trajectory. It continues along this trajectory until x^v has moved forward by 30cm or it senses an opposing force of 30N along the X-axis. If this force limit is exceeded, the

robot assumes that either the twisting controller was unsuccessful or an obstruction is blocking the door.

After completing this push, the robot once again uses its laser range finder to see if the door has moved. If the door did not move, the twisting controller failed on that tip of the handle and the robot tries to manipulate the other tip. If twisting both the tips of the handle is unsuccessful, the robot reports that the door is locked.

5.5 Evaluation of Controllers for Twisting Door Handles and Pushing Open Doors

In section 5.5.1, we report on 157 trials with a single door, and empirically show that our method of twisting door handles is robust with respect to the orientation of the virtual trajectory, the stiffness of the arm, and the friction between the end effector and the handle. Then in section 5.5.2 on page 76, we present the performance of the individual components and the integrated system on 8 doors in 43 trials.

5.5.1 Twisting Door Handles

To provide intuition about the mechanics of door handle twisting, fig. 52 on the following page shows a simple quasistatic model of the interaction forces between the end effector of the manipulator and a door handle. If we model the door handle as a pin joint with a torsional spring of stiffness K and ignore the mass of the door handle, then

$$N = K\theta/l \tag{27}$$

$$f \leq \mu_s N \quad \text{(no slip)} \tag{28}$$

$$f = \mu_k N \quad \text{(slip)} \tag{29}$$

where N is the normal force, θ is the angle of the door handle with the horizontal, l is the distance between the point of contact and the axis of the handle, f is the friction force and μ_s and μ_k are the coefficients of static and kinetic friction. The normal force increases as the door handle is twisted, and the direction of friction force changes depending on the direction of impending (or actual) slip.

Friction is important for our handle twisting behavior as it prevents the end effector from slipping off (or slipping up) the handle. If a robot rigidly grasps the door handle, it can be modeled as a high or infinite coefficient of friction. If the end effector motion corresponds exactly to the geometry of the door handle, there will be no impending slip and the interaction force will be as small as possible for a fixed contact location on the handle and equal to the normal force. With zero

friction, the end effector will only sense the normal force. Our previous work relied on quasistatic motion and low friction to estimate the normal force and push against it [90].

Figure 52: ***Left:*** *Model of the end effector twisting a door handle.* ***Middle:*** *In plane forces on the end effector from the handle.* ***Right:*** *Figure showing how the handle is constrained to move (red), the different lines along which we varied the virtual trajectory for the end effector, and the initial position of the end effector when the twisting behavior starts (black). The length of the handle is 12cm and the end effector made contact approximately 4cm from the tip of the handle. The angle of the door handle at which the door can be pushed open is shown as a dashed red line.*

We now describe experiments on one door handle to evaluate the performance of the twisting controller (section 5.4.4.2 on page 71) when its parameters are varied.

5.5.1.1 Angle of the Virtual Trajectory

While twisting the handle, the Cartesian-space virtual trajectory is the sum of a line parallel to the surface of the door and small motions normal to the surface (see section 5.4.4.2 on page 71). In this section we empirically show that the performance of the system is robust with respect to the orientation of the line.

To simulate variation due to navigation, we put the robot in three different orientations in front of the door (-10°, 0° and 10°) with the handle in the workspace of the manipulator. For each orientation, we varied the angle of the line in increments of 20° from -40° to 70° as shown in fig. 52 and performed five trials for each angle. We kept all the other control parameters constant.

The robot successfully twisted the handle and pushed the door in 15 out of 15 trials when the angle of the line was between -20° and 40°, failed three times out of 15 when the angle was 60°, and was unsuccessful in all trials for -40° and 70° (see fig. 53 on the next page). For -40° the end effector slipped off the door handle and for 70° it slipped up the handle towards the pin joint. This suggests that there is a range of orientations of the virtual trajectory line for which our control strategy will be successful.

Figure 53 also shows the magnitude of the maximum interaction force between the end effector and the handle and door for the different trajectories. This magnitude includes the component of the interaction force normal to the surface of the door. From the model of the door handle and figs. 52 and 53, we see that as the line deviates from the trajectory of a point on the handle, the maximum

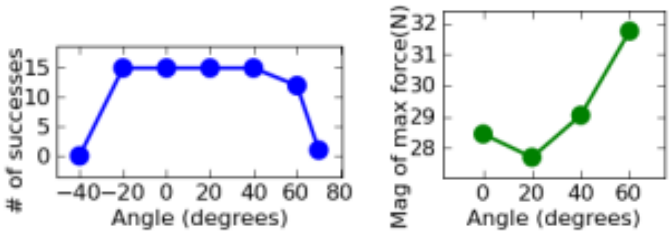


Figure 53: ***Left:*** *Number of successes (out of 15) for different angles for the virtual trajectory line of the twisting behavior.* ***Right:*** *Average (over 15 trials) of the magnitude of the maximum force measured by the wrist force-torque sensor for different angles of the virtual trajectory line.*

force measured at the end effector increases. Given our model, we expect that this results from increased frictional forces and a reduced moment arm.

Minimizing the interaction forces during manipulation reduces the chance of damage. Based on the results shown in fig. 53, 20° appears to reduce the interaction forces and sits near the middle of the successful range of angles. Unless otherwise noted, we use 20° for the trajectory angle of twisting in the remaining experiments.

5.5.1.2 Stiffness of the Manipulator

In this section we show how the maximum interaction force between the handle and the manipulator varies as the stiffness of the manipulator changes. The joint stiffness settings that we used for all the experiments result in a stiffness at the end effector of approximately $1800N/m$ normal to the surface of the door, and $250N/m$ and $200N/m$ in the horizontal and vertical directions parallel to the plane of the door. The stiffness is almost equal in the plane of the door and much higher in the direction normal to the surface of the door.

In this experiment we varied the stiffness of the manipulator by scaling the joint-space stiffness and damping matrices, K_j and D_j (see fig. 45 on page 66), by a scalar α. We set α equal to 1.0 for all experiments other than this one. For this experiment, we varied α from 0.6 to 1.2.

As shown in fig. 54 on the next page, the robot successfully twisted the handle and pushed the door open in five out of five trials for each value of α. At the lower values, 0.6 and 0.8, the end effector had a tendency to slip on the handle, but still managed to succeed at the task. When we increased α to 1.2, the magnitude of the maximum interaction force increased. As the stiffness increases, the arm should behave more like a position controlled arm, and we would expect greater interaction forces.

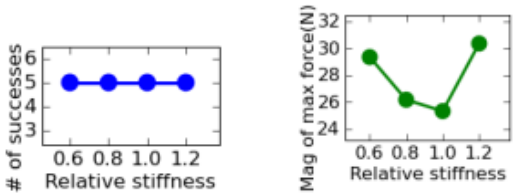
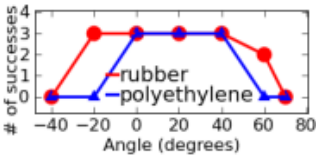


Figure 54: ***Left:*** *Number of successes (out of 5) for different stiffness settings.* ***Right:*** *Average (over 5 trials) of the magnitude of the maximum force measured by the wrist force-torque sensor while twisting the handle for different stiffness settings of the virtual joint springs.*

5.5.1.3 Friction

In all the other experiments, we covered the end effector with a sheet of rubber to increase the friction between the end effector and the door handle. We now show how the performance of the handle twisting behavior changes with variation in the friction. In this experiment we covered the end effector with polyethylene which has a lower coefficient of friction on steel than rubber (see fig. 55) [209]. In practice, we would also expect a change in the contact properties due to variations across door handles.



Coefficient of friction

	Static	Kinetic
rubber-steel	0.6-0.9	0.3-0.6
polyethylene-steel	0.2	

Figure 55: ***Top:*** *Number of times (out of three) the robot successfully twisted the handle and pushed open the door with different materials on the end effector (rubber and polyethylene) as a function of the angle of the virtual trajectory.* ***Bottom:*** *Coefficient of static and kinetic friction for rubber-steel and polyethylene-steel contact [209].*

Figure 55 shows the results of repeating the experiments of section 5.5.1.1 on page 74 with rubber and polyethylene on the end effector. When polyethylene was in contact with the handle, the end effector slipped more easily. The range of angles for the virtual trajectory for which door handle twisting was successful was smaller, but our control strategy was successful in pushing open the door with both types of contact.

5.5.2 The Complete System

We carried out a total of 43 door opening trials on 8 different doors, shown in fig. 56. In each trial, the robot started out approximately 1.5m perpendicular to the door and less than 0.5m to the right or left of the door handle in the direction parallel to the door. The starting orientation of the robot was either approximately perpendicular to the door or facing towards the door handle (a variation of

around 30°). We clicked on a point in the laser scan to select a location in the world (as explained in section 5.4.1 on page 66). The robot's task was to navigate to the door handle, twist it and push open the door.

Figure 56: ***Top:*** *The door handles of the 8 doors used in the experiments.* ***Middle:*** *One of the five starting locations for each of the 8 doors that we tested our system with.* ***Bottom:*** *Pictures of the robot after it has successfully twisted the door handle and pushed open the door by a small angle. For the results we report, the doors are numbered 1 to 8 from left to right.*

Table 6 on the following page shows the performance of the individual components on the 8 different doors. We tested each component four times and the preceding components were successfully executed prior to each test. For these experiments, 'Servo to handle' was deemed successful if the robot servoed to the door such that the door handle was within the workspace of the manipulator. 'Segment and touch handle' required the robot to segment the door handle and make contact with it. 'Twist and push' was successful if the robot successfully twisted the door handle and pushed open the door. 'Determine if locked' required the robot to correctly report whether the door was locked or not. Failure to open an unlocked door also results in a failure of 'Determine if locked' because the robot incorrectly reports the door to be locked.

Our implementation of manipulation with compliant trajectories performed well in these tests. The robot successfully twisted the door handle in 31 out of 32 trials (96.9%). The complete system including perception and navigation succeeded with unlocked doors in 28 out of 32 trials (87.5%) and locked doors in 8 out of 8 trials (100%). There were two failures in 'Servo to handle', one failure in segmenting the handle, and one failure in twisting the handle.

Our method of servoing to the door handle relies on detecting the handle in every planar scan

Table 6: *Performance of the component behaviors on 8 different doors.*

Door #	Door state	Servo to handle	Segment and touch handle	Twist and push	Determine if locked
1	Unlocked	4/4	4/4	4/4	4/4
	Locked	1/1	1/1		1/1
2	Unlocked	3/4	4/4	4/4	4/4
	Locked	1/1	1/1		1/1
3	Unlocked	4/4	4/4	4/4	4/4
	Locked	1/1	1/1		1/1
4	Unlocked	4/4	4/4	3/4	3/4
	Locked	1/1	1/1		1/1
5	Unlocked	4/4	4/4	4/4	4/4
	Locked	1/1	1/1		1/1
6	Unlocked	4/4	3/4	4/4	4/4
	Locked	1/1	1/1		1/1
7	Unlocked	3/4	4/4	4/4	4/4
	Locked	1/1	1/1		1/1
8	Unlocked	4/4	4/4	4/4	4/4
	Locked	1/1	1/1		1/1
Success Rate		38/40	39/40	31/32	39/40
Success %		95.0%	97.5%	96.9%	97.5%

from the laser range finder. Thin door handles, small errors in the estimated height of the handle, and motion of the sensor while the robot moves can result in the robot losing track of the door handle. We believe that more sophisticated methods for servoing and reliable odometry would reduce the chance of the robot failing to navigate to the door handle.

The segmentation failure was due to incorrect estimation of the orientation of the door from the Hough transform (section 5.4.2.1 on page 68). The robot failed to twist the door handle once when the end effector slipped towards the axis of rotation of the handle, and the twisting force exceeded its threshold even though the door was unlocked.

5.6 Pulling Open Doors with a Stationary Mobile Base

In this section we present two controllers to pull open doors and drawers without prior knowledge of the mechanism kinematics. The input to our system is a rough location for the handle of the mechanism, and an initial orientation for its hook end effector (fig. 57). We also present a method to estimate the kinematics of the mechanism while the robot is opening it. In section 5.7 on page 85, we extend the second controller described in this section to coordinate an omnidirectional base and

the compliant arm while pulling open doors and drawers.

Figure 57: ***Left:*** *Examples of a human using his hand as a hook.* ***Right:*** *The two orientations for the robot's hook, Left and Up.*

We designed the hook, printed it with a 3D printer, and then applied rubber to its surfaces to increase friction. One can think of this as a model of the human hand when a person uses a finger or fingers to hook around a handle and pull something open. We also took inspiration from prosthetic hooks, which have been successfully used with remarkable versatility and effectiveness. A hook has the advantage of being effective for a variety of handles, including recessed handles that would be difficult to grasp.

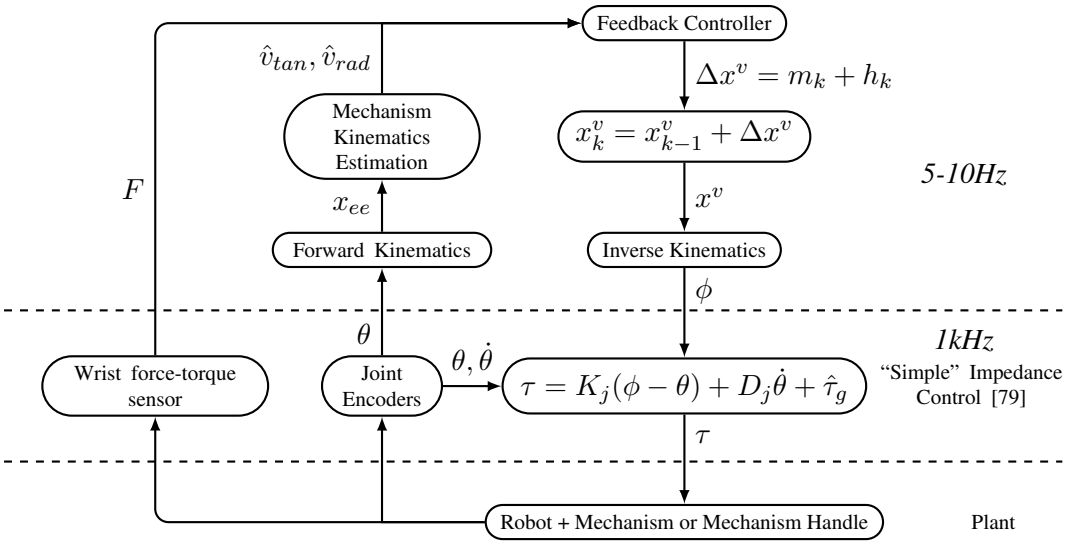


Figure 58: *Block diagram of the different components of our system for pulling open mechanisms with a compliant arm and a stationary base. The controller frequencies are specific to our implementation.*

Figure 58 on the current page shows the different components of the overall system. We have written the feedback controllers in Python and they run on a Mac Mini running Ubuntu GNU/Linux. At approximately 10Hz, these controllers compute an incremental change in the Cartesian-space virtual trajectory (Δx^v) using 6-axis force feedback from the wrist-mounted force-torque sensor (F), and estimates of the mechanism kinematics using the history of end effector positions (x_{ee}).

5.6.1 Two Feedback Controllers for Pulling Open Doors and Drawers

We now describe two controllers that generate Cartesian-space virtual trajectories that enable our robot to pull open novel doors and drawers. Both controllers share a similar structure:

$$\Delta x^v = m(k) + h(k) \tag{30}$$

$$x^v(k) = x^v(k-1) + \Delta x^v \tag{31}$$

At each time step, k, they compute an incremental change in the Cartesian-space virtual trajectory, Δx^v, by adding a vector intended to operate the mechanism, $m(k)$, and a vector intended to keep the hook from slipping off of the handle, $h(k)$.

$x^v(-1)$ is determined by the controller that attempts to get a firm hook on the handle (section 5.6.3.1 on page 83). As a result, there is an initial offset between the Cartesian-space virtual trajectory and the actual position of the end effector, for example in fig. 59 on the next page.

5.6.1.1 Stopping Conditions

The robot looks for three types of stop conditions. First, if the magnitude of the force measured by the wrist force-torque sensor exceeds a maximum force threshold $F_{th}(k)$, the robot stops. While operating the mechanism, the robot computes $F_{th}(k)$ as

$$F_{th}(k) = \begin{cases} 80N & k \leq k_{mv} \\ min(\|F[k_{mv}]\| + 30N, 80N) & \text{otherwise} \end{cases} \tag{32}$$

$$k_{mv} = min\{k \text{ s.t. } \|x_{ee}(k) - x_{ee}(0)\| \geq 10cm\} \tag{33}$$

where k_{mv} is the time step when the tip of the hook end effector has moved by a distance greater than 10cm, and $F(k_{mv})$ is the measured force at time t_{mv}. $F_{th}(k)$ is initialized to 80N, and is adapted to be 30N greater than $F(k_{mv})$ after time k_{mv}. This addresses the higher initial forces that are often required when opening doors and drawers. In chapter 7 on page 116, we use data-driven models of forces while pulling open doors to compute $F_{th}(k)$ and show that this enables the robot to detect certain collisions faster and with lower force applied to the mechanism.

Second, if the magnitude of the force drops below 1N for more than one second, the robot assumes that its hook end effector has slipped off the handle and stops. Finally, if the Cartesian-space virtual trajectory leaves the workspace of the arm, the robot stops. Under some circumstances, the virtual trajectory could be allowed to leave the workspace, but we do not address these situations in this work.

5.6.1.2 Controller 1: Pull-Linear

Equations (34) and (35) show the two vectors $m(k)$ and $h(k)$ for this controller. $m(k)$ is a constant vector which results in a linear Cartesian-space virtual trajectory and moves x^v towards the robot's torso by 1cm at each time step. So,

$$m(k) = (-1cm, 0, 0) \tag{34}$$

$$h(k) = (0, 0, 0). \tag{35}$$

An example of the resulting linear Cartesian-space virtual trajectory is shown in fig. 59.

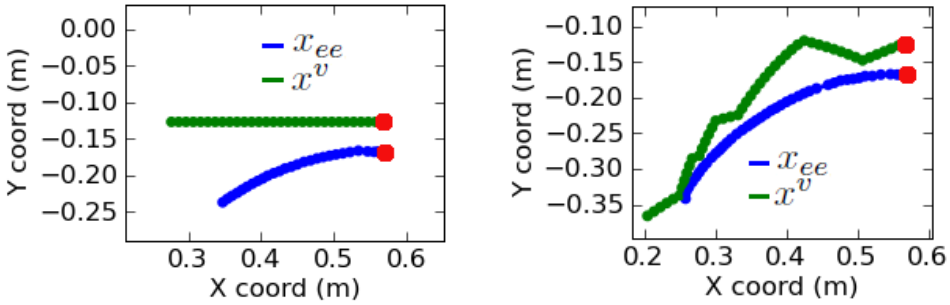


Figure 59: *Figure showing the Cartesian-space virtual trajectory in green and the actual motion of the end effector in blue for the Pull-Linear **(left)** and Pull-Radial-Force **(right)** controllers. The red circles denote x^v and the end effector position at $k = -1$.*

5.6.1.3 Controller 2: Pull-Radial-Force

This controller assumes that the mechanism will either be a prismatic joint or a rotary joint with the axis of rotation parallel to gravity. It uses an estimate of the location of the axis of rotation and the radius of the mechanism using the algorithm detailed in section 5.6.2 on the next page.

Based on this estimate, the controller defines tangential and radial unit vectors for the estimated rotary motion, $(\hat{v}_{tan}(k), \hat{v}_{rad}(k))$. This is comparable to estimating a task frame [165, 133, 29]. Starting at $k = 0$, and until the mechanism kinematics estimation algorithm of section 5.6.2 does not have enough points to estimate the kinematics, the Pull-Radial-Force controller sets these vectors to $((-1, 0, 0), (0, 1, 0))$.

Using the tangential and radial unit vectors, the controller factors the force measured by the wrist force torque sensor into estimated tangential and radial components, $(\hat{F}_{tan}(k), \hat{F}_{rad}(k))$.

$m(k)$ is a vector of constant magnitude oriented in the direction of the estimated motion tangent (see eq. (36)). By itself, $m(k)$ would tend to create a Cartesian-space virtual trajectory that looks similar to the trajectory traced out by the handle of the mechanism. $h(k)$ is a vector of constant

magnitude that is parallel to the radial unit vector, $\hat{v}_{rad}(k)$. The sign of the vector is determined by a bang-bang control law that attempts to keep the radial force applied to the handle by the hook at 5N, as shown in eq. (37). The controller is:

$$m(k) = 1cm \cdot \hat{v}_{tan}(k) \tag{36}$$

$$h(k) = \begin{cases} -0.25cm \cdot \hat{v}_{rad}(k) & \text{if } \hat{F}_{rad}(k) < 5N \\ +0.25cm \cdot \hat{v}_{rad}(k) & \text{if } \hat{F}_{rad}(k) > 5N \\ (0,0,0) & \text{otherwise} \end{cases} \tag{37}$$

An example of the resulting virtual trajectory is shown in fig. 59 on the preceding page.

5.6.2 Mechanism Kinematics Estimation

The mechanism kinematics estimation algorithm returns an estimate of the location of the axis of rotation of the mechanism that the robot is operating and its radius. It initially computes two models using the end effector trajectory in the XY plane (via forward kinematics). One model is for rotary mechanisms that open to the right and the second for rotary mechanisms that open to the left. The algorithm does not have a separate model for prismatic joints and we assume that a prismatic joint will be estimated as a rotary joint with a very large radius. The algorithm runs an optimization for each model to compute:

$$(c, r)^* \quad = \quad argmin(\sum_k (\|x_{ee}(k) - c\| - r)^2) \tag{38}$$

where r is the radius of the mechanism, c is the location of its axis in the XY plane, and $x_{ee}(k)$ is the end effector trajectory. The initial guess for r is set to 1.0m and the initial guess for c is ($x_{ee}(0)_x$, $-r$) and ($x_{ee}(0)_x$, r) for optimizations for door models that open to the right and left respectively. $x_{ee}(0)_x$ is the x coordinate of the end effector at $k = 0$.

We use an implementation of the BFGS optimization algorithm from SciPy [102]. Out of the two circle models, the kinematics estimation algorithm selects the model with the lower residual error as the current estimate of the mechanism's kinematics. This gives new estimates at approximately 5Hz.

5.6.3 Sources of Variation

The two pulling controllers depend on several factors: the way in which the handle has been hooked, the stiffness settings of the joints, the initial posture of the arm, and the pose of the body relative to the handle. Within this subsection, we describe how we selected these parameters.

5.6.3.1 A Behavior for Hooking the Handle

First, performance depends on the way in which the robot has hooked the handle. For example, if the end effector slips off the handle, or along the handle, the performance of the system will degrade. To represent the natural variations that might occur when autonomously hooking a handle, we created a hooking behavior.

Since we only use haptic sensing in this work, we do not address the problem of finding a good place to hook onto a door or drawer. Instead, we provide the robot with a 3D location to which the hook should be moved and the orientation for the hook (see fig. 57 on page 79).

The robot hooks on to the handles through two compliant motions. First, it moves the arm with a linear Cartesian-space virtual trajectory towards the mechanism until it detects contact with the surface of mechanism using the wrist force-torque sensor at a point on the surface offset from the handle location. The robot then moves the hook laterally toward the handle along the surface of the mechanism. It uses a bang-bang controller that at each time step moves the virtual trajectory out of the surface by 0.2cm if the pushing force exceeds 3N, and into the door by 0.2cm if the force falls below 1N. Pushing against the surface allows the robot to hook onto recessed handles.

The robot continues the lateral motion along the surface until a force threshold indicates contact, or the hook has moved a distance of 10cm. These two motions are illustrated in fig. 60.

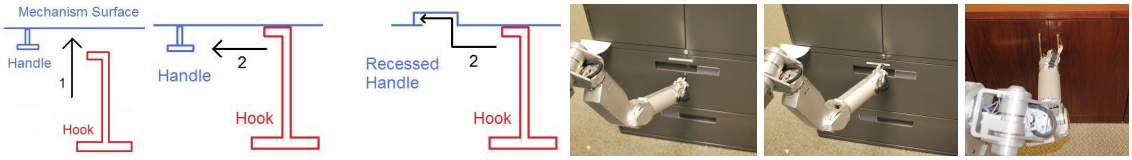


Figure 60: ***(a)-(b):*** *Pictorial representation of the two motions to hook onto handles.* ***(c)-(e):*** *This method works for both recessed and protruding handles.*

5.6.3.2 Selecting the Stiffness Values

For all of the experiments, we used the same relative stiffnesses of the five joints, and only altered a common scaling factor α. With α set to 1.0 the stiffnesses for the shoulder flexion/extension, shoulder abduction/adduction, shoulder internal/external rotation, elbow flexion/extension, and forearm pronation/supination motions at 20, 50, 15, 25, and 2.5 Nm/rad, respectively. The joint stiffness values are also similar to measured stiffnesses of humans during planar reaching motions [194]. We use position control for the wrist flexion/extension and abduction/adduction, since the passive

compliance of the SEA springs and the cables connecting the SEAs to these two joints introduce significant compliance. These values are the same as chapter 3 on page 24.

5.6.3.3 The Pose of the Body Relative to the Handle

During autonomous activities, the position of the base and torso relative to the handle will be likely to vary due to uncertainties (e.g., perception), limited precision (e.g., motion of the base), task constraints (e.g., obstacles), and other challenges that accompany real-world operation. We wish to verify empirically that our proposed controllers are robust to these forms of variation.

Height of the Torso: Qualitatively, we found that system performance was not sensitive to the height of the torso relative to the handle. Consequently, we chose to fix the height of the torso relative to the handle for a given hook orientation. We chose this height by searching for a value that would maximize the area of the planar Cartesian workspace of the end effector. We estimated the size of the workspace by using the IK solver to sample over achievable end effector positions as shown in fig. 61. The robot attempts to keep the height of the torso relative to the handle as close to the height with the maximum estimated area as possible, subject to the joint limits of the linear actuator.

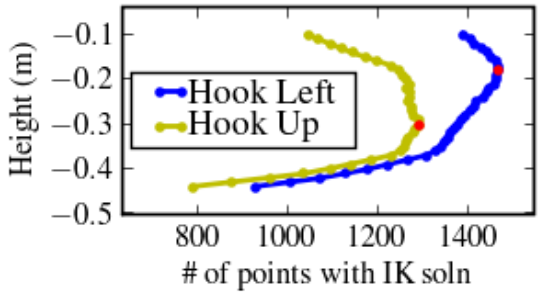


Figure 61: *This figure shows the workspace in planes parallel to the ground, estimated as the number of points in a grid of 1cm resolution that have an inverse kinematics solution. The red dots denote the heights with the maximum number of points (-0.3m and -0.18m).*

Planar Location and Orientation: To test the performance of the controllers with variation in the position and orientation of the base relative to the handle of the mechanism we carried out multiple trials on the three mechanisms. We varied the angle over 30° (±15° of error) and the distance by 10cm, as shown in fig. 66 on page 92. The results of these trials are detailed in section 5.8.1 on page 91.

In a second set of trials, we selected the position and orientation of the robot from the first set of trials that resulted in the maximum opening angle and carried out one trial each on 12 different

mechanisms. Section 5.8.2 on page 93 presents the performance of the robot in these trials.

For all the trials, the robot initially pulled towards its torso, as though it believed the initial motion should be normal to its torso.

5.6.3.4 The Initial Posture of the Arm

Given the fixed pose of the wrist, the height of the torso, and the planar position and orientation of the base, the arm must reach the handle such that it is hooking it. For our tests, this leaves one DoF remaining in the 7DoF arm. We initialize the posture of the arm such that the plane formed by the shoulder, elbow and wrist is close to vertical with the elbow tilting away from the torso.

5.7 Coordinating an Omni-directional Base and Compliant Arms

In this section we present an extension to the Pull-Radial-Force controller described in section 5.6 on page 78 that enables the robot to move its omnidirectional base while pulling open the door. This results in an increased effective workspace for the compliant arm.

We use this controller as part of an intergrated system that lets a user click on the handle in an image from the robot's cameras, and the robot then navigates to the mechanism and pulls open the door or drawer without prior knowledge of the mechanism kinematics.

5.7.1 Integrated System Description

Figure 62 on the current page shows the three main actions that the robot executes in this paper to pull open doors and drawers. We do not address the problem of finding a good place to hook onto a door or drawer in this work. Instead, we provide the robot with a 3D location to which the hook should be moved, the direction in which the hook should be oriented. The hook can be in one of two orientations, upward and to the left, as shown in fig. 57 on page 79. In the experiments with this controller, the starting orientation of the robot is normal to the plane of the front of the door or drawer.

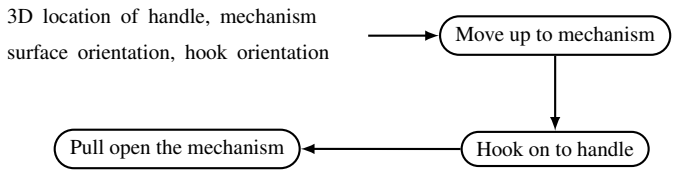


Figure 62: *Input to the integrated system and the three main actions that the robot executes in to pull open doors and drawers.*

5.7.1.1 The User Interface

For an assistive application, we have previously shown interfaces by which a person can designate a 3D location in the world using a laser pointer and a touch screen [35]. For this work, the robot uses a calibrated camera registered with a tilting laser range finder to present the user with a point-and-click interface with which the user can select the location of a handle to be manipulated. The user first uses a mouse to click on a handle in the image. The robot then estimates the 3D position associated with this clicked location using the registered 3D point cloud from the tilting laser range finder. It simply selects the 3D point that is closest to the ray associated with the user selected pixel. The user also specifies the direction in which the handle should be hooked.

5.7.1.2 Hooking on to Handles

After the user provides the requisite information, the robot navigates to the mechanism and aligns itself parallel to the surface using visual odometry from a camera looking down at the floor [111].

The robot updates its estimate of the location of the handle as it navigates close to it. Once the handle is within the workspace of the arm, it attempts to firmly hook the handle before executing the pull behavior using the behavior described in section 5.6.3.1 on page 83.

It then runs a controller to pull open the mechanism, which we describe in the rest of this section.

5.7.1.3 Components of Subsystem for Pulling Open Mechanisms

Figure 63 on the following page shows the different components of our subsystem for pulling open mechanisms once the robot has hooked on to the handle.

Feedback Controller: This is the main control block for the subsystem. It is responsible for adjusting the Cartesian-space virtual trajectory of the compliant arm. When the virtual trajectory is at risk of falling outside of the arm's workspace, the controller commands the base to move the robot such that x^v will move towards the interior of the arm's workspace. The inner workings of this block are described in more detail in sections 5.7.2 and 5.7.3 on pages 87–89. This controller runs at approximately 10Hz.

Mechanism Kinematics Estimation: This block monitors the trajectory of the tip of the robot's hook in the world frame, and uses this information to infer the kinematics of the mechanism that the robot is operating. It assumes that the mechanism will either be a rotary joint or a prismatic joint whose trajectory lies in a plane parallel to the floor. It fits a circle to the trajectory of the hook end effector to estimate the location of the axis of rotation, as described in section 5.6.2 on page 82. It

uses forward kinematics (x_{ee}) and visual odometry ($\hat{o}$) to estimate trajectory of the tip of the hook in the fixed world frame. It runs at approximately 5Hz.

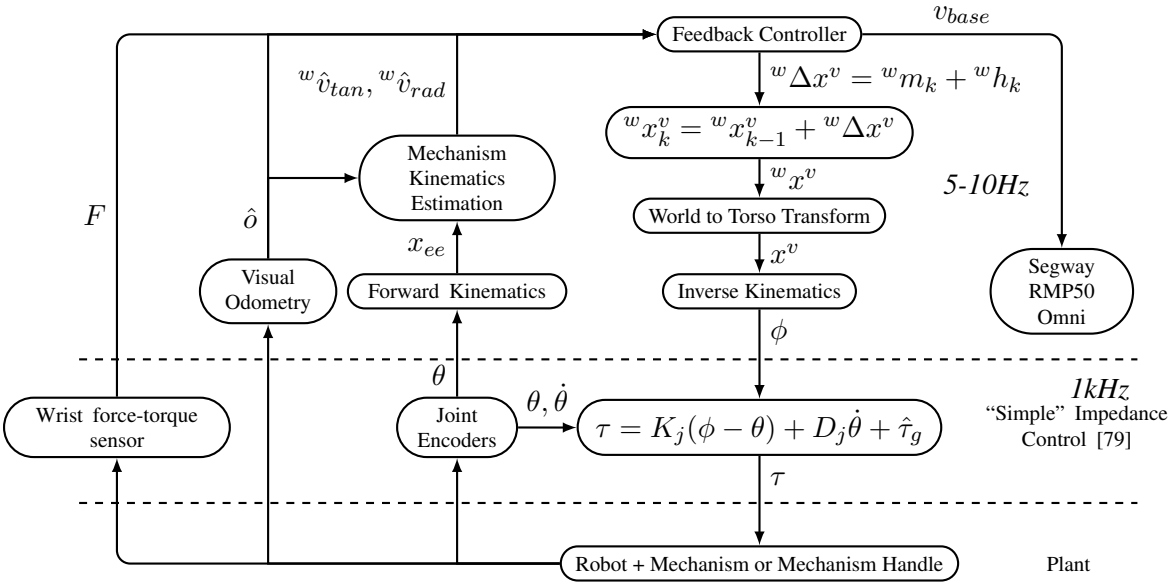


Figure 63: *Block diagram of the different components of our subsystem for coordinating an omnidirectional mobile base and a compliant arm while pulling open mechanisms. The controller frequencies are specific to our implementation.*

Visual Odometry: The robot uses a floor-facing camera placed underneath the mobile base with a light ring to compute visual odometry [111]. As it moves, the robot tracks visual features across sequential images of the floor. Due to wheel slip, we found odometry based on the wheel encoders to be poor. Visual odometry estimates the position and orientation of the mobile base, $\hat{o}$, at approximately 10Hz.

Compliant Arm Control: Our inverse kinematics computation and the 1kHz "simple" impedance control are described in section 5.3 on page 65.

Segway RMP50 Omni-directional Mobile Base: The omnidirectional mobile base accepts velocity commands, v_{base}, from the feedback controller.

F/T sensor: The force-torque sensor is at the base of the hook end effector (see fig. 57 on page 79), on the wrist of the robot's arm and outputs 6-axis force-torque measurements.

5.7.2 Controlling the Compliant Arm to Pull Open Doors and Drawers

In this subsection, we describe how the feedback controller of fig. 63 generates Cartesian-space virtual trajectories for the compliant arm in a fixed world frame that enable the robot to pull open novel doors and drawers. The fixed world frame, denoted with the left superscript w, is attached to

the initial location of the handle. The X axis of this coordinate frame is normal to the surface of the mechanism and points away from the robot, and the Z axis points vertically up. This is in contrast to the controllers described in section 5.6.1 on page 80 in which the virtual trajectory was specified with respect to a frame of reference attached to the robot's torso. By specifying the virtual trajectory with respect to the world frame, we can consistently use it even when the robot moves its base.

At each time step, k, the controller computes an incremental change to the Cartesian-space virtual trajectory in the world frame, ${}^{w}\Delta x^{v}(k)$, by adding a vector intended to operate the mechanism, ${}^{w}m(k)$, and a vector intended to keep the hook from slipping off of the handle, ${}^{w}h(k)$. So,

$$ {}^{w}\Delta x^{v}(k) = {}^{w}m(k) + {}^{w}h(k) \tag{39} $$

$$ {}^{w}x^{v}(k) = {}^{w}x^{v}(k-1) + {}^{w}\Delta x^{v}(k) \tag{40} $$

where the left superscript w denotes that all the coordinates are in the fixed world frame. Section 5.7.2.1 details how the vectors ${}^{w}m(k)$ and ${}^{w}h(k)$ are computed.

${}^{w}x^{v}(k)$ is then transformed into an equilibrium point in the local coordinate frame of the torso, $x^{v}(k)$, using

$$ x^{v}(k) = T_{w}(\hat{o}(k-1)) \cdot {}^{w}x^{v}(k), \tag{41} $$

where T_w is the matrix that transforms points from the world coordinate frame to the local coordinate frame of the torso. T_w is parameterized by the translation and rotation of the omnidirectional base estimated by visual odometry, $\hat{o}(k-1)$.

5.7.2.1 Operating the Mechanism While Staying Hooked Onto the Handle

The mechanism kinematics estimator estimates tangential and radial unit vectors $({}^{w}\hat{v}_{tan}, {}^{w}\hat{v}_{rad})$ for the trajectory of the handle. This is comparable to estimating a task frame [165, 133, 29]. Starting at $k = 0$, until the mechanism kinematics estimator has enough points to estimate the kinematics, the controller updates ${}^{w}x^{v}$ in a linear trajectory towards the robot.

Using the radial and tangential unit vectors, the controller factors the force measured by the wrist force-torque sensor into estimated tangential and radial components, $(\hat{F}_{tan}(k), \hat{F}_{rad}(k))$.

${}^{w}m(k)$ (see eq. (39) on the current page), a vector of constant magnitude oriented in the direction of the estimated motion tangent, is calculated as

$$ {}^{w}m(k) = 1cm \cdot {}^{w}\hat{v}_{tan}(k) \tag{42} $$

By itself, ${}^{w}m(k)$ would tend to create a CEP trajectory that looks similar to the trajectory traced out by the handle of the mechanism. To prevent the hook from slipping off the handle, the controller computes ${}^{w}h(k)$ (see eq. (39)) as

$$ {}^{w}h(k) = 0.1cm/N \cdot (\hat{F}_{rad}(k) - 5N) \cdot {}^{w}\hat{v}_{rad}(k) \tag{43} $$

Hence, ${}^{w}h$ is a vector, parallel to the radial unit vector (${}^{w}\hat{v}_{rad}$), whose length is determined by a proportional controller that attempts to keep the radial force applied to the handle by the hook at 5N.

In addition, the robot tries to keep the orientation of the hook aligned with the estimated task frame subject to the joint limits of the arm. The goal of keeping the hook aligned with the task frame is to reduce the chance of the hook slipping off while operating the mechanism.

5.7.2.2 Stop Conditions

The robot looks for three types of stop conditions, as described in section 5.6.1.1 on page 80.

5.7.3 Moving the Omni-Directional Base

We now describe how the feedback controller of fig. 63 on page 87 commands velocities for the omnidirectional base with the goal of keeping the virtual trajectory and the tip of the robot's hook in the interior of the arm's workspace.

Let $Bndry$ be the set of points on the boundary of the planar Cartesian workspace of the arm. The controller first selects either x_{ee} or x^v, whichever is closer to the boundary, by computing

$$ x_{close} = \underset{x \,\epsilon\, \{x^v, x_{ee}\}}{argmin} \; (distance(x, Bndry)) \tag{44} $$

where x^v is the Cartersian-space virtual trajectory of the end effector and x_{ee} denotes the coordinates of the robot's end effector. All the the coordinates are in the local frame of the torso.

To determine the direction in which to move the omnidirectional base, we use a reactive control technique that has been used for controlling mobile robot bases [18]. We compute the sum of "repulsive forces", r, on x_{close} as

$$ r = \sum_{p \epsilon C} \frac{(x_{close} - p)}{\|x_{close} - p\|^2} \tag{45} $$

$$ C = \{p \,\epsilon\, Bndry \; s.t. \; \|x_{close} - p\| < 10cm\} \tag{46} $$

where C is the set of points in $Bndry$ that are less than $10cm$ from x^l_{close}.

The controller commands the base to move at a speed of 15cm/s opposite to the resultant vector (r) by setting the commanded base velocity (v_{base}) to

$$v_{base} = -\frac{r}{\|r\|} \cdot 15cm/s \tag{47}$$

If x_{close} is greater than 5cm from $Bndry$, the controller sets $v_{base} = 0$. Figure 64 shows the computed base motion direction for two cases: $x_{close} = x^v$ and $x_{close} = x_{ee}$.

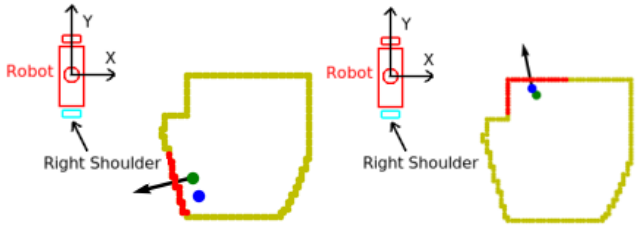


Figure 64: *Figure illustrating the computed direction of motion for the omnidirectional base (black arrow) to move the virtual trajectory (green) or the tip of the hook (blue) away from the boundary of the planar workspace of the arm (yellow). The red points are those that belong to the set C of eq.* (46)*, and are used to compute the direction of motion for the mobile base.*

5.7.4 Coordinating the Omni-directional Base and the Compliant Arm

Equation (39) on page 88 updates the Cartesian-space virtual trajectory in a coordinate frame attached to the initial location of the handle and fixed in the world. Equation (41) on page 88 then transforms ${}^w x^v$ into the local frame of the torso using the position and orientation of the mobile base, thereby coordinating the motion of the mobile base and the arm.

If the distance of x_{close} (eq. (44) on the preceding page) from the boundary of the planar workspace of the arm falls below 3cm, we modify eq. (39) to be

$${}^w \Delta x^v(k) = {}^w h(k) \tag{48}$$

i.e. we set ${}^w m(k)$, the vector intended to operate the mechanism, to zero if x_{close} is near the workspace boundary. We now allow for the motion of the base to move the virtual trajectory and the tip of the hook away from the workspace boundary.

5.7.5 Sources of Variation

The pulling controller depends on several factors: the way in which the handle has been hooked, the stiffness settings of the joints, the initial posture of the arm, and the pose of the body relative to the handle. The robot selects these parameters as described in section 5.6.3 on page 82.

5.8 Evaluation of Controllers for Pulling Open Doors

We now present our evaluation of the controllers for pulling open doors with a stationary base in sections 5.8.1 and 5.8.2 on pages 91–93, and the extension that enables to robot to coordinate its omnidirectional base and the compliant arm while performing this task in section 5.8.3 on page 95.

5.8.1 Testing The Two Controllers on Three Mechanisms

To test the performance of the Pull-Linear and Pull-Radial-Force controllers (described in section 5.6.1 on page 80) with variation in the position and orientation of the base relative to the handle of the mechanism and with variation in the stiffness of the manipulator, we evaluated the two controllers when operating the left and right doors of a cabinet with rotary joints, and a drawer with a prismatic joint, shown in fig. 65. During these tests, we varied the pose of the robot's base relative to the handle and the overall stiffness of the arm, as described next.

Figure 65: *The three mechanisms used for the experiments of section 5.8.1.1.* ***Left:****Two cabinet doors with recessed handles.* ***Right:*** *Drawer with a recessed handle.*

5.8.1.1 Comparison of Pulling Controllers

In this subsection, we report on 108 trials that we conducted to empirically compare the performance of the Pull-Linear and Pull-Radial-Force controllers. For these trials, we systematically varied the mechanism being operated, the pose of the base, and the overall stiffness of the arm. Similar to the experiments of section 5.5.1.2 on page 75, we varied the stiffness of the manipulator by scaling the joint-space stiffness and damping matrices, K_j and D_j (see fig. 45 on page 66), by a scalar α.

For each of the three mechanisms, we tested both controllers with the base in six different poses and the arm's stiffness scaled by three different values of α (i.e., $3 \times 2 \times 6 \times 3 = 108$ trials). Figure 65 shows the three mechanisms and fig. 66 on the following page shows the six poses of the base. We set α to 0.8, 1.0, and 1.2.

Table 7 on the next page summarizes the performance of the two pulling controllers subject to these variations. Both Pull-Linear and Pull-Radial-Force opened all three mechanisms through

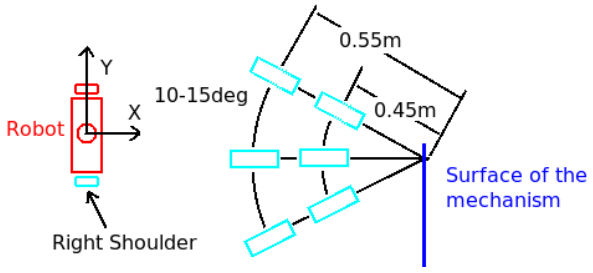


Figure 66: *We placed the robot in approximately the 6 different positions shown in the figure with an angle between -15° and +15° with the surface of the mechanism and the base of the arm being either 0.45m or 0.55m from the handle.*

an average angle greater than 30° for the rotary joints and an average distance greater than 20cm for the prismatic joint. Also, the robot correctly distinguished between rotary and prismatic joints and estimated the correct direction of rotation with both of the controllers. For this experiment, we interpreted a radius greater than 2m to be a prismatic joint.

Table 7: *Comparison of the two pulling controllers.*

Mechanism		Pull-Linear	Pull-Radial-Force
Door 1 (Opens right)	Avg max force	20.5N	9.0N
	Std max force	4.9N	1.8N
	Max angle opened	41.6°	79.0°
	Avg angle opened	32.8°	47.2°
	Std angle opened	4.8°	16.4°
	Avg estimated radius	0.331m	0.357m
	Std estimate radius	0.028m	0.018m
Door 2 (Opens left)	Avg max force	9.5N	7.6N
	Std max force	4.0N	3.0N
	Max angle opened	43.4°	42.0°
	Avg angle opened	31.2°	30.6°
	Std angle opened	7.4°	4.2°
	Avg estimated radius	0.680m	0.600m
	Std estimate radius	0.397m	0.322m
Drawer	Avg max force	15.8N	14.1N
	Std max force	2.3N	1.4N
	Max distance opened	0.290m	0.340m
	Avg distance opened	0.223m	0.243m
	Std distance opened	0.051m	0.056m

For the door that opens to the right, both of the control methods obtained a good estimate of the radius of the joint with the average estimated radius being 0.33m for Pull-Linear (13.2% relative error) and 0.357m for Pull-Radial-Force (6% relative error). The radius of the cabinet door,

measured by hand, is 0.38m. The error and the standard deviation of the estimated radius for Door 2 (which opens to the left) are quite high. This is because the end effector tended to slip off the handle. As a result, the trajectory from forward kinematics was not indicative of the trajectory of the handle.

The average of the maximum interaction force is higher for Pull-Linear as compared to Pull-Radial-Force. This is expected since Pull-Radial-Force actively adjusts the virtual trajectory via a bang-bang controller to keep $\hat{F}_{rad}$, the component of the force along the radial direction, at 5N. In these experiments, the mobile base was stationary while the robot manipulates the mechanism. Due to this, and the workspace limits of the robot arm, the position of the robot relative to the mechanism had a significant impact on the angle through which the robot opened a door, and the distance through which it pulled a drawer. The effect of base position on the extent to which the robot opened a mechanism is shown in fig. 67 on this page.

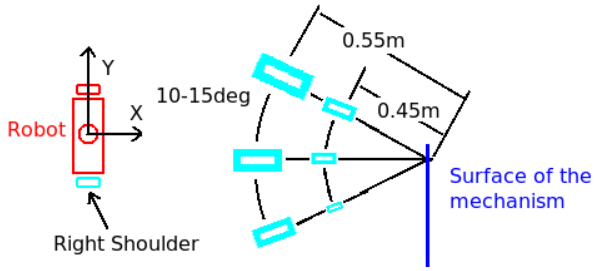


Figure 67: Figure showing the variation of the opening angle for a rotary joint with the position of the robot. The larger the size of the light blue rectangle, the greater the angle through which the robot opened the door for the experiments of section 5.8.1.1.

The performance of the two controllers was similar for the prismatic joint. Scaling the stiffness and damping to 0.8, 1.0, and 1.2 did not result in a significant change in the task performance.

5.8.2 Testing One Controller on 12 Mechanisms

For this experiment, we used the Pull-Radial-Force controller on 12 different mechanisms: four drawers, five doors that open to the right, and three doors that open to the left. We selected the position of the robot relative to the mechanism that resulted in the maximum opening angle for the rotary joints in the experiments of the previous section (see fig. 67). We fixed the arm posture, oriented the hook, and then drove the base and elevated the torso with a gamepad such that the hook was poised to haptically reach forward and hook the handle.

Table 8 on the next page presents the performance of the robot on each mechanism and fig. 68 shows the robot after it has operated each of the 12 mechanisms. The order of the mechanisms in

table 8 and fig. 68 is consistent. The image of the drawer without the robot is the only mechanism that the robot failed to open. While attempting to pull open this drawer, the hook slipped off before the drawer opened. This drawer initially requires a very large force to open ($> 50N$), which suggests that $h(k)$ should apply higher forces to keep the hook on the handle. The robot correctly distinguished between prismatic and rotary joints and the direction of rotation in the remaining 11 trials.

Figure 68: *Images showing the robot after it has operated 11 mechanisms with the Pull-Radial-Force controller. The image of the drawer without the robot is the mechanism for which the hook slipped off before the drawer opened.*

Table 8: *Performance of Pull-Radial-Force on 12 mechanisms.*

Mechanism	Angle/Distance pulled	Estimated Radius	Measured Radius
Right Door 1	39.0°	0.51m	0.57m
Right Door 2	84.6°	0.30m	0.34m
Right Door 3	84.4°	0.37m	0.40m
Right Door 4	20.0°	0.67m	0.80m
Right Door 5	50.0°	0.89m	0.17m-0.44m
Left Door 1	54.1°	0.42m	0.17m-0.44m
Left Door 2	30.0°	0.60m	0.57m
Left Door 3	38.1°	0.54m	0.41m
Drawer 1	0.3m	45.63m	N/A
Drawer 2	Failed	Failed	N/A
Drawer 3	0.3m	9.69m	N/A
Drawer 4	0.3m	7.34m	N/A

Figure 69 shows the relative error for each of the eight rotary mechanisms. There were larger errors in the mechanism kinematics estimation when the robot operated the mechanisms through smaller angles (such as Right Door 4), or if the hook slid along the handle (such as Right Door 5 and Left Door 1). In our experiments, small opening angles occurred either on doors that opened to the left or on doors with large radii. Currently, the robot only uses its right arm and keeps its mobile base stationary. As a result, it can open doors that open to the left and doors with large radii through a smaller angle as compared to doors that open to the right.

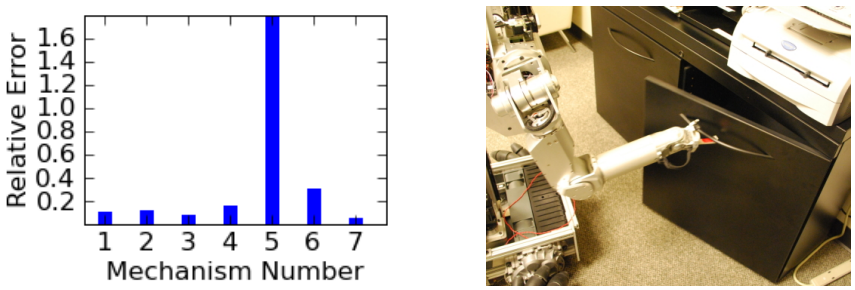


Figure 69: ***Left:*** *Relative error in the radius estimation for the eight rotary mechanisms.* ***Right:*** *A wide horizontal handle, coupled with rotary motion in the XY plane resulted in the hook slipping along the handle as the robot opened this cabinet. As a result, the trajectory from forward kinematics was not informative about the radius of the mechanism and the location of its axis.*

The robot estimated the radius of the rotary joints with an average relative error of 16.3%. This average error ignores a misestimation of the radius for the cabinet shown in fig. 69. This unusual cabinet has its handle horizontally oriented on top, which gives a lot of room for the hook to slide along the handle in the XY plane. When opening the cabinet, the hook slid along the handle, which our current methods for kinematic estimation do not accommodate. Consequently, although the robot correctly categorized the door as being a rotary joint that opens to the right, the robot misestimated the radius as being 89cm, while hand measurement reported it to be between 17cm and 44cm.

5.8.3 Coordinating an Omni-directional Base and Compliant Arm

We now present our evaluation of the integrated system described in section 5.7 on page 85 that enables the robot to move its mobile base while pulling open doors and drawers, thereby increasing the effective workspace of the arm. We evaluated the performance of the system on 10 different mechanisms: four cabinet doors that open to the right, three cabinet doors that open to the left, and three drawers.

We performed four trials for each mechanism. The robot started around 1m from the location

of the handle, aligned with the surface of the mechanism. The task for the robot was to navigate up to the mechanism and operate it, as shown in fig. 70. We deemed a trial to be successful if the robot navigated to the mechanism and opened it through an angle greater than 60° for rotary mechanisms or 30cm for linear mechanisms. We measured the angle using a protractor that we stuck to the mechanism which can be seen in the images in fig. 72 on the next page. The input to the robot was the 3D location of the handle and an orientation for the hook.

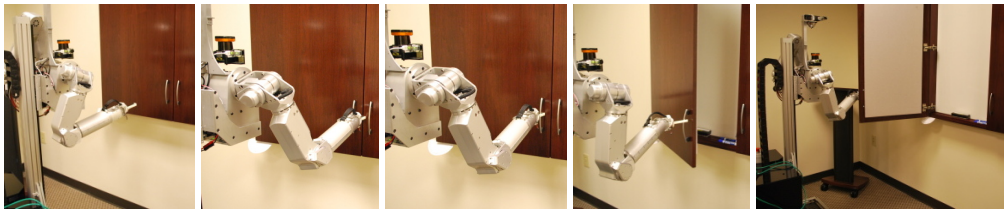


Figure 70: *Sequence of images that show the robot opening a door. The robot moves up to the mechanism, reaches out with the arm to make contact with the surface, moves the hook along the surface to hook onto the handle and then pulls open the door, moving the omnidirectional base to increase the workspace of the robot.*

To verify that the manipulation behavior is robust to the position of the robot, the approach behavior moved the robot to a different target position in each of the four trials. Figure 71 on the following page shows the four different target positions of the robot relative to the handle after navigating to the handle and prior to the start of the manipulation behavior. We varied the relative target position of the robot by 20cm parallel to the surface of the mechanism and 10cm normal to the mechanism. Ideally, if the robot navigated precisely to target position 2 in fig. 71, the selected handle location would be in the middle of the arm's planar workspace. The remaining three target positions were defined relative to target position 2. We began each of the four trials on one mechanism with the robot in approximately the same starting position.

Figure 72 shows the robot after it has pulled open each of the 10 mechanisms in one of the trials and table 9 on page 98 summarizes the performance of the robot. The order of mechanisms in fig. 72 and table 9 correspond. The robot successfully opened rotary mechanisms through an angle greater than 60° in 26 out of 28 trials and pulled the linear mechanism through a distance greater than 30cm in 11 out of 12 trials. Overall, on 10 different mechanisms, the robot was successful in 37 out of 40 trials.

Two of the three failures were due to the robot not being able to hook onto the handle. We believe that this was due to a combination of errors in the odometry and the initial estimate of the location of the handle. The third failure occurred on Right Door 2, which has a large handle along

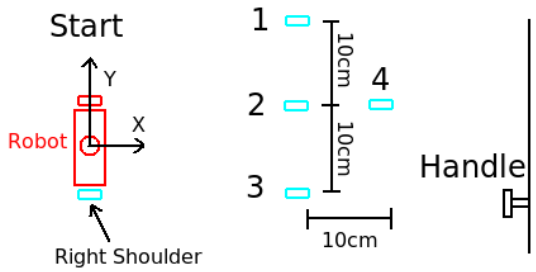


Figure 71: *This figure shows the approximate starting position of the robot and the four different target positions of the robot relative to the selected location of the handle after navigating to the handle and prior to the start of the manipulation behavior for the experiments. Ideally, if the robot navigated precisely to target position 2, the selected handle location would be in the middle of the arm's planar workspace. The remaining three target positions are defined relative to target position 2.*

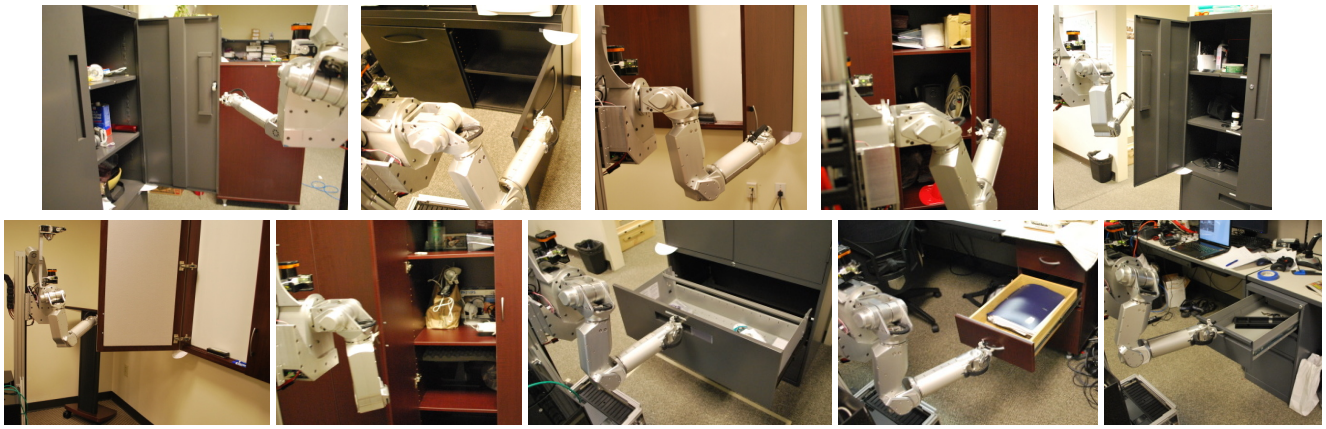


Figure 72: *Images showing the robot after it has operated the 10 mechanisms of the evaluation of Section 5.8.3. The order of images is consistent with the results of table 9 on the next page.*

which the hook can slip easily. The robot opened Right Door 2 through an angle of 40° and the pulling behavior stopped due to the force threshold being exceeded. In addition to this task failure, the hook slipping along the handle of Right Door 2 resulted in large errors in the robot's estimate of the kinematics for this mechanism, see table 9.

5.9 Conclusion

In this chapter we presented two different integrated systems that enable a robot with compliant arms and force sensing at the end effector to open doors and drawers. The robot did not have prior information about the mechanism type (e.g, rotary or prismatic), its kinematics (e.g, the radius of the door), or the forces required to open the mechanism. The input to our systems was a 3D location of the door handle and an orientation for the robot's hook end effector.

We empirically demonstrated that this level of input was sufficient for our robot to open doors

Table 9: *Performance of the robot on 10 mechanisms.*

Mechanism	Angle/Distance pulled		Estimated Radius		Measured Radius	Max Magnitude of Total Resultant Force		Success Rate
	mean	std	mean	std		mean	std	
Right Door 1	85.0°	11.2°	0.37m	0.01m	0.39m	25.8N	3.6N	4/4
Right Door 2	67.5°	17.9°	1.77m	1.63m	0.17m-0.44m	33.5N	4.7N	3/4
Right Door 3	75.0°	3.5°	0.58m	0.03m	0.57m	19.7N	1.1N	4/4
Right Door 4	85.0°	8.7°	0.4m	0.04m	0.42m	23.9N	3.0N	4/4
Left Door 1	83.8°	8.2°	0.33m	0.02m	0.34m	34.3N	3.0N	4/4
Left Door 2	103.8°	8.2°	0.47m	0.02m	0.57m	30.9N	2.9N	4/4
Left Door 3	93.3°	2.4°	0.33m	0.05m	0.41m	34.7N	3.2N	3/4
Drawer 1	0.44m	0.0m				38.4N	2.0N	4/4
Drawer 2	0.48m	0.0m				37.1N	4.3N	3/4
Drawer 3	0.39m	0.0m				39.6N	3.1N	4/4

and drawers with a high success rate. We also demonstrated that our controllers can be used to coordinate the motion of a robot's omnidirectional base and compliant arm while pulling open doors and drawers. While pulling open doors and drawers, the robot was also able to make online estimates the kinematics of the mechanisms.

Although the systems that we presented in this chapter robustly opened a variety of doors and drawers, they did not have any expectations about the forces involved in these manipulation tasks. As such, the robot could pull with a force up to 80N before deciding that a door is locked or that a collision occurred. This is significantly higher than the force required to open most kitchen cabinets ($\sim$5N) but is comparable to the force required to open spring loaded doors. As we show in the next two chapters (chapters 6 and 7 on pages 99–116), by capturing and modeling the forces during a manipulation task we can give robots common sense about everyday forces, and enable them to detect anomalous conditions faster and with lower excess force.

CHAPTER VI

CAPTURING THE FORCES AND TRAJECTORIES THAT OPEN DOORS AND DRAWERS

Little is known about the statistics of real-world mechanical systems in human environments nor the implications of these statistics for robot design. In this chapter, we describe progress towards capturing and characterizing real-world mechanical systems that are relevant to assistive robots. With the example of doors and drawers, we show that even ostensibly simple real-world devices relevant to instrumental activities of daily living (IADLs) have complex structure that can be captured and characterized (see fig. 73 on the following page). We also present evidence that this structure can be used to inform the design of both hardware and software for assistive mobile manipulators.

As described in section 5.1 on page 64, researchers have developed a number of robotic systems to operate doors between rooms, and open cabinet doors and drawers. Other work in service robotics has used observations to estimate kinematic parameters of doors and drawers and articulated rigid bodies [94, 206, 106]. Robots have also estimated mechanical parameters to perform tasks, such as friction coefficients when pushing objects [201].

These efforts have focused on algorithms for control and estimation, not on capturing the statistics of everyday mechanics. Their empirical evaluation has been limited to a relatively small number of mechanisms. Despite progress towards assistive robots capable of opening doors and drawers, the answers to basic questions have been unclear, such as, "How hard does a robot need to pull to open most doors?" and "How high does a robot need to reach to open most drawers?".

We describe our efforts to address these questions by capturing kinematic trajectories and forces while operating 29 doors and 15 drawers in 6 homes and one office building in Atlanta, GA, USA. We have also hand-measured the kinematics of 299 doors and 152 drawers in 11 area homes. The work presented in this chapter has previously appeared in the proceedings of the *IEEE RAS/EMBS International Conference on Biomedical Robotics and Biomechatronics* [96].

We have organized the rest of this chapter as follows. We describe our capture setup and capture procedure in section 6.1 on page 101, and present simple mechanical models for doors and drawers in section 6.3 on page 104. We then analyze the kinematic and force data in section 6.4 on page 107. In section 6.5 on page 111, we give examples of how our captured data could be used to inform design decisions for assistive robots. We then conclude with sections 6.6 and 6.7 on pages 114–115.

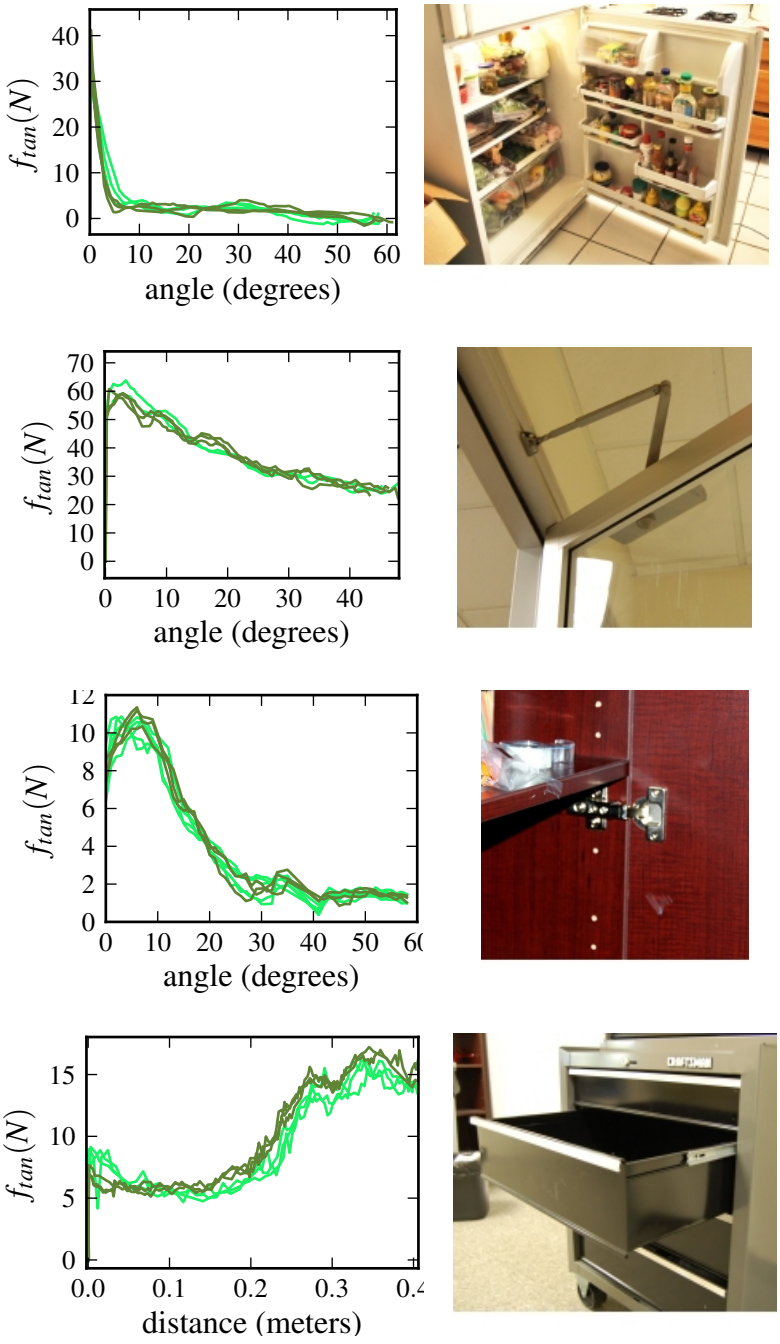


Figure 73: *Forces recorded while opening four mechanisms. Left plots show forces tangential, f_{tan}, to the motion of the handle as a function of the device's configuration. Lighter green indicates trials with higher average velocity. Pictures on the right highlight a key mechanical element of each mechanism.* ***Top:*** *refrigerator, 6 recordings, avg. velocities of 17.8°/s to 26.8°/s. High initial force due to low pressure interior.* ***Upper Middle:*** *springloaded door, 5 recordings, avg. velocities of 6.5°/s to 13.5°/s. Large forces throughout movement due to linkage at top.* ***Lower Middle:*** *kitchen cabinet, 9 recordings, avg. velocities of 7.4°/s to 16.3°/s. Non-linear spring keeps it closed with max force at about 4°.* ***Bottom:*** *toolchest drawer, 6 recordings, avg. velocities of 0.07m/s to 0.15m/s. Larger force halfway due to 2nd stage of telescoping rail.*

6.1 Methodology

6.1.1 Force and Motion Capture Setup

We designed a custom force and motion capture system to log the kinematic trajectory through which mechanisms move and the forces used to operate them. Figure 74 shows the components of our capture system which consists of: (1) a checkerboard pattern that we attach to the mechanism (2) a handheld end-effector that consists of a hook, a force-torque sensor (ATI Nano25), and four checkerboard patterns (3) a camera (Point Grey DragonFly2) to capture video of the operator (4) studio lights (5) a backpack containing hardware for the force-torque sensors, and (6) a laptop to log the data. The handheld end-effector consists of a 3D printed hook with a force-torque sensor at its base, four laser cut acrylic pads with checkerboard patterns, and a handle for the operator to grasp. The four checkerboard patterns allow us to ensure that one checkerboard is visible given typical hook orientations.

We have used open source code for this capture setup, including ROS [170], OpenCV [7], and have released our code and hardware designs (see appendix A.4 on page 147).

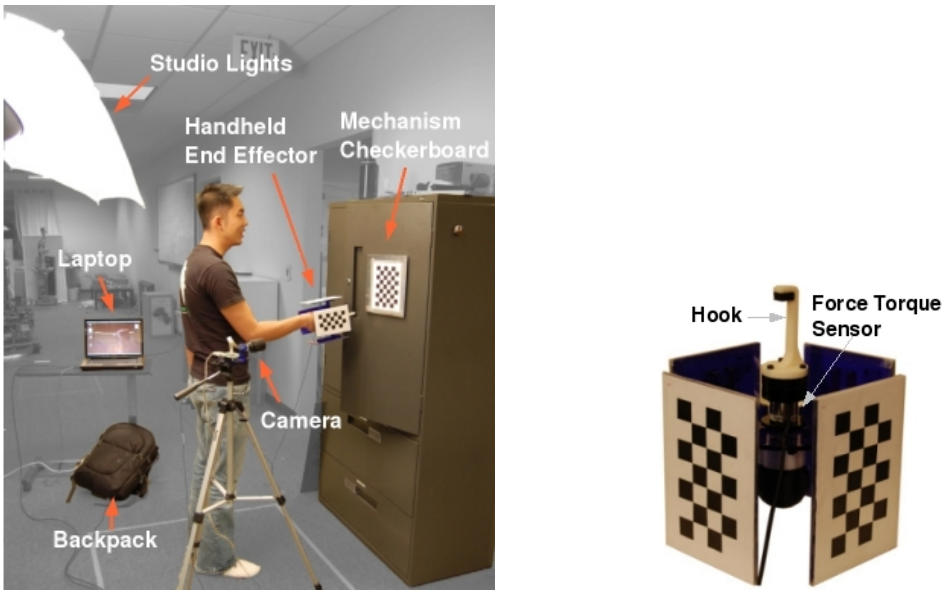


Figure 74: ***Left:*** *The different components of our capture setup.* ***Right:*** *The handheld end effector consists of a 3D printed hook with a force-torque sensor at its base and four checkerboard patterns.*

6.1.2 Capture Process

Research in the field of haptics, has attempted to capture and model everyday mechanics to produce realistic haptic simulations in virtual environments [51, 187, 129, 17, 217]. This body of work

emphasizes high-fidelity models of single objects for realistic haptic feedback.

In contrast, we wish to capture statistics that can inform the design of assistive robots. Since robots operating at moderate to slow speeds would have value, we create quasi-static models and characterize the statistics of real-world mechanisms. We capture data relevant to quasi-static models by instructing the operators to open the doors and drawers at slow speeds. We then throw out recordings of trials with large average velocities. For each mechanism we first find the lowest average velocity across all the trials. We then throw out all trials whose average velocity is larger than this lowest average velocity by more than 10°/s or 15cm/s.

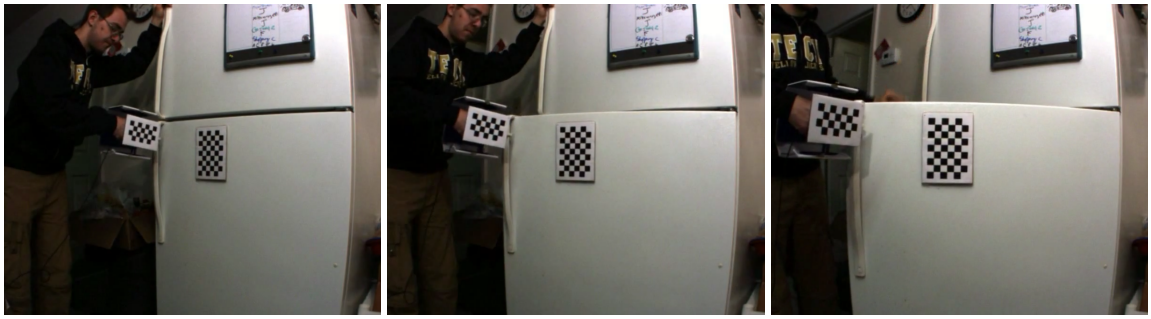


Figure 75: *Sample images recorded with the force and motion capture system while a human operator was opening a refrigerator door. A video of a single trial of data capture on four different mechanisms: a refrigerator, a spring loaded door, a drawer, and a kitchen cabinet is part of the supplementary materials (appendix A.4 on page 147).*

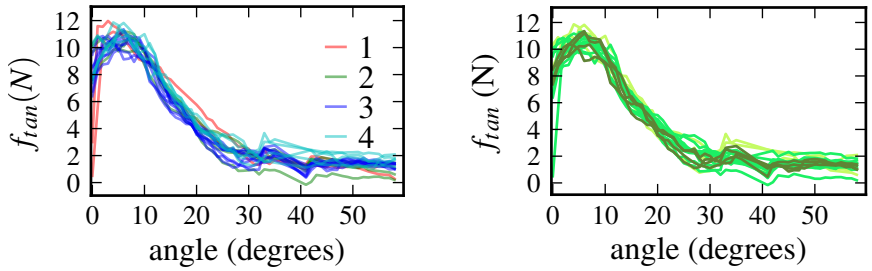


Figure 76: *Forces recorded when four different operators opened the same mechanism. The forces were dominated by the configuration of the mechanism and not by the operator.* ***Left:*** *Trials colored by the operator.* ***Right:*** *Trials colored by average velocity. Lighter green indicates trials with higher average velocity.*

To capture data, we use the handheld hook to pull the door or drawer open and record force-torque sensor readings and video. Figure 75 shows three images from the camera of our data capture system while an operator was opening a refrigerator. The operator for any mechanism in our dataset was one of four different people. We also conducted trials for four operators on one mechanism and found that the measured force exhibited no clear variation due to the user and was strongly

dependant on the quasi-static forces due to the configuration of the mechanism (see fig. 76 on the current page).

6.1.3 Time Synchronization

We log the data from the force-torque sensors at around 100Hz, and video from the camera at 30 frames per second. To synchronize the two data streams we smooth the force data independently for each channel, and resample at 33Hz.

6.1.4 Mechanisms that we surveyed

We collected data in six homes and one office environment in Atlanta, Georgia, USA. At each location, we tried to capture data for all the different kinds of mechanisms subject to the constraints of our capture setup. First, we were unable to collect data for mechanisms whose trajectories are not parallel to the ground (e.g. dishwashers) or would occlude the checkerboard patterns. Second, we were limited to mechanisms that can be operated by hooking then pulling. This excludes all the doors between rooms at the locations we visited, except two spring loaded doors in the office environment. Third, drawers in the kitchens of two of the six houses had a tendency to fall down as they are pulled out (fig. 82 on page 111). We did not capture data for these drawers.

If there were multiple mechanisms with the same hinge type, we collected data for one of them. For example, we collected data for one drawer out of a set of identical looking drawers. We logged 10 trials for each mechanism.

In addition, we hand-measured kinematic properties of every mechanism in 11 different homes (a total of 451 doors and drawers). We measured the height of the handle of the mechanism above the ground and the distance of the hinge of the mechanism from the hooking location (for rotary joints), and the opening distance for drawers.

6.2 Estimating Kinematic Parameters

We use the recorded video to estimate the configuration of the mechanism that the human operator opened and a task fram at the handle of the mechanism.

6.2.1 Estimating the Mechanism Configuration

The configuration of a mechanism, which we denote as q corresponds to the opening angle for a door, and the distance by which a drawer has been pulled open. For each frame of the captured video, we estimate the 6D pose of the mechanism checkerboard pattern in the coordinate frame of

the camera with the [42]. We then fit either a rotary or a linear model to this trajectory using [206] and use that to compute the configuration q of the mechanism for each frame of the video. We assume that the mechanism is closed at the start of the data capture ($q = 0$ for the first frame of the video).

6.2.2 Estimating a Task Frame at the Handle

As detailed in section 6.3, we factor the forces measured using the force-torque sensor into a component that is responsible for opening the mechanism and a second component that produces only constraint forces. To do this, we estimate a task frame at the point of contact between the handheld end-effector and the mechanism handle [132].

The orientation of the task frame for a rotary mechanism is shown in fig. 77b on the current page. It is important to note that if the handle sticks out from the surface of a rotary mechanism, the orientation of this task frame will be different from the orientation of the coordinate frame of the mechanism checkerboard pattern. We fit a circle to the trajectory of the point on the handheld end-effector that makes contact with the mechanism handle. We assume that this point is fixed relative to the hook. We then use the radial direction of the circle for our estimate of x^{task}, and the normal to the plane of the circle as the estimate of y^{task}. We compute z^{task} as $x^{task} \times y^{task}$.

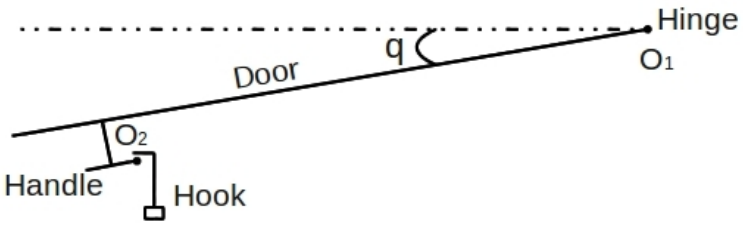
For a linear mechanism, we assume that this task frame is the same as the coordinate frame of the mechanism checkerboard pattern. This is the same as assuming that the mechanism checkerboard pattern is mounted such that the normal to the surface of the pattern and the direction of motion of the drawer are identical.

6.3 Measuring the Opening Forces

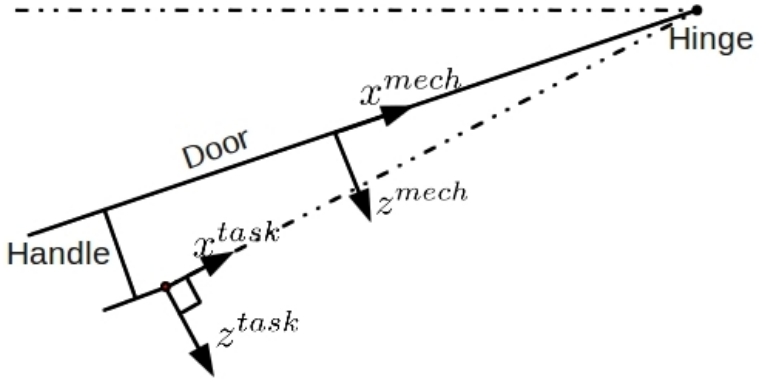
We would like to use the captured data to compute statistics that will be useful for the design of robots. In this section we model drawers as prismatic joints and doors as rotary joints, and use basic mechanics to estimate the forces that open the doors and drawers in our dataset. These models can be violated in the real world. For example, there can be some amount of motion in the joint of a door and some drawers have a tendency to fall down as they are pulled out (fig. 82 on page 111).

6.3.1 Notation

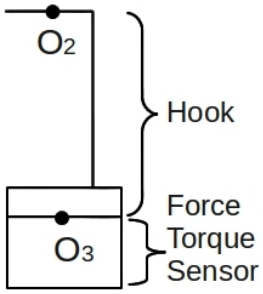
We use the tuple $W(A) = (f(A), m(A))$ to denote the force $f(A)$ and moment $m(A)$ due to wrench W at a point A on a rigid body. The value of the wrench W at a different point B on the



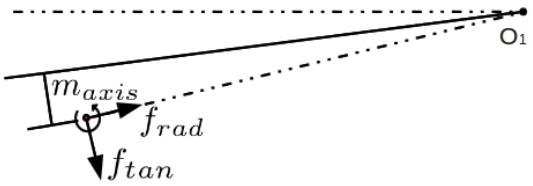
(a) Top view of a mechanism as we use a hook to open it. O_1 is a point on the axis of rotation of the mechanism. O_2 is the point of contact between the hook and the handle.



(b) The orientation of the task frame at the handle can be different from the mechanism coordinate frame.



(c) Hook with the force-torque sensor at its base. The force-torque sensor measures the wrench at point O_3.



(d) Components of the wrench $W^m(O_2)$ computed using eq. (50) on page 106. f_{rad} and f_{tan} are the components of the force in the plane of the trajectory of the hook and m_{axis} is the moment parallel to the axis of the rotary mechanism.

Figure 77: *Diagrams illustrating the kinematics and forces while opening a rotary joint with an instrumented hook. We use this rotary joint model for doors to estimate the kinematics of the mechanism (section 6.2 on page 103) and the forces required to open doors (section 6.3 on the preceding page).*

same rigid body can be computed as

$$\begin{pmatrix} f(B) \\ m(B) \end{pmatrix} = \begin{pmatrix} f(A) \\ m(A) + P(BA) \times f(A) \end{pmatrix}, \tag{49}$$

where $P(BA)$ is the vector from point B to point A.

6.3.2 Mechanics of a Rotary Joint

Figure 77a on the next page shows a rotary mechanism with a hinge at point O_1, opened by an angle q using the hook which makes contact with the handle at point O_2. Let the wrench applied to the force-torque sensor by the hook measured at the base of the hook be $W^s(O_3)$. Using eq. (49), we can compute the value of W^s at the point of contact between the hook and the handle of the mechanism O_2, denoted by $W^s(O_2)$. Now, let W^m be the wrench applied to the mechanism by the hook. Assuming that the mass and moments of inertia of the hook are small enough for us to ignore dynamic effects, we get

$$-W^m(O_2) - W^s(O_2) + W^g(O_2) = 0, \tag{50}$$

where W^g is the wrench due to gravity on the hook. At the start of each trial, when we know that W^m is equal to zero, we compute W^g by measuring W^s from the force torque sensor. We then assume that W^g is constant for the duration of the trial, although the hook orientation might change.

Equation (50) allows us to compute the wrench applied on the mechanism by the hook, W^m. We then compute the forces f_{tan} and f_{rad} and the moment m_{axis}, shown in fig. 77d on the preceding page. For our capture setup, m_{axis} is almost zero indicating that contact between the hook and the handle of the mechanism can be approximated by a pin joint. We assume a pin joint model for the point of contact and set m_{axis} equal to zero.

To compute f_{tan} and f_{rad}, we estimate a task frame at O_2, see section 6.2.2 on page 104. f_{rad} points in the radial direction and will only result in constraint forces at the hinge of the mechanism. f_{tan} is the only component of the force that contributes to a moment about the mechanism axis. The moment about the hinge axis due to W^m will be equal to $f_{tan} \cdot r$, where r is the distance of O_2 from the mechanism axis.

6.3.3 Mechanics of a Linear Joint

We model drawers as a prismatic joint whose direction of motion is along z^{task}, as explained in section 6.2.2 on page 104. Using this model, the only component of the wrench W^m that is responsible

for opening the drawer is the force along z^{task}. All the other components will result in constraint forces.

6.4 Statistics of Doors and Drawers

We now present statistics of the forces and kinematics from our dataset of measurements from doors and drawers. We used the force and motion capture system described in section 6.1 on page 101 to recording the kinematic trajectories and forces while opening 29 doors and 15 drawers in 6 homes and one office building in Atlanta, GA, USA. We also hand-measured the kinematics of 299 doors and 152 drawers in 11 area homes.

6.4.1 Kinematic Analysis

In this section we provide some examples of statistics from our hand-measured dataset. We discuss the implications of these and similar data for robot design in section 6.5 on page 111.

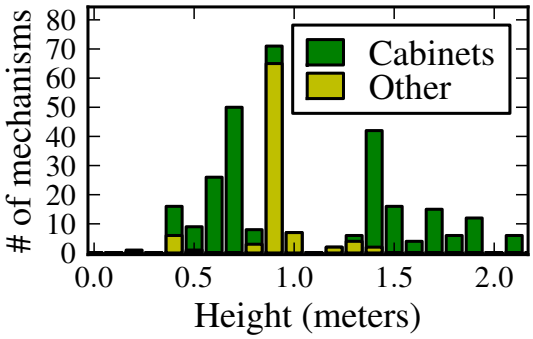
Rotary Mechanisms: Figure 78a on the current page shows the distribution over heights of the door handles. We divide this distribution into two with the first containing the handle heights of cabinets, and the second containing the handle heights of all other mechanisms.

There are few cabinet handles between a height of 0.9m and 1.3m. Often in kitchens in the US, countertops are 91.4cm (36") high, and cabinets sit 18" above countertops or 124.2cm above the ground [1]. The most frequent non-cabinet rotary mechanisms are doors between rooms. The standard height for the handles for these doors, 91.4cm (36"), corresponds to the peak in the yellow histogram in fig. 78a. Other doors such as storage and coat closet doors occur at lower frequencies. In fig. 78b, we also show the distribution of the radii, split into cabinets (green) and all other mechanisms (yellow). Cabinets come in a variety of standard sizes starting at 22.5cm (9"), increasing in increments of 3" up to 122cm (48") [2].

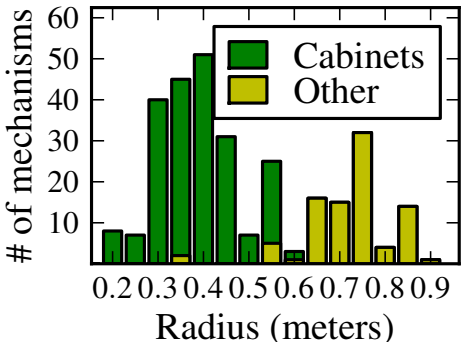
Drawers: From fig. 79a we observe that most drawer handles are less than 1m above the ground. Thus, if a mechanism handle is at a height greater than 1m, there is a strong prior that the mechanism is not a drawer. Alternately, if a robot has a specific controller or behavior for opening drawers, it might be sufficient for that behavior to work at a maximum height of 1m above the ground.

6.4.2 Force Analysis

As detailed in section 6.1.2 on page 101, we capture data relevant to quasi-static models by instructing the operators to open the doors and drawers at slow speeds and then throw out recordings of trials with large average velocities. Figure 73 on page 100 shows that there are only modest changes

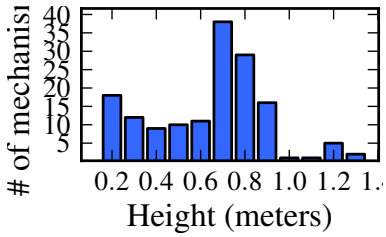


(a) Histograms of height of the handles of doors. Each bin represents a range of 10cm.

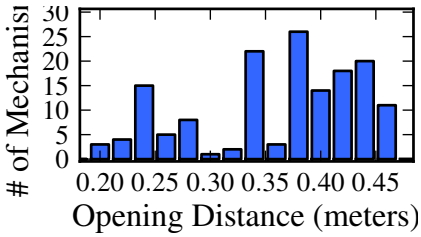


(b) Distributions of door radii with each bin having a range of 5cm.

Figure 78: *Histograms of kinematic properties of rotary mechanisms (doors) from our hand-measured kinematic dataset of 299 doors. Green represents cabinets and yellow represents other mechanism classes (doors between rooms, freezers, and refrigerators).*



(a) Histogram of height of drawer handles above the ground. Each bin has a range of 10cm.

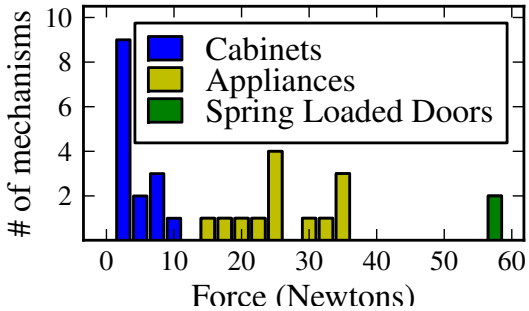


(b) Histogram of maximum opening distance for drawers. Each bin has a range of 2cm.

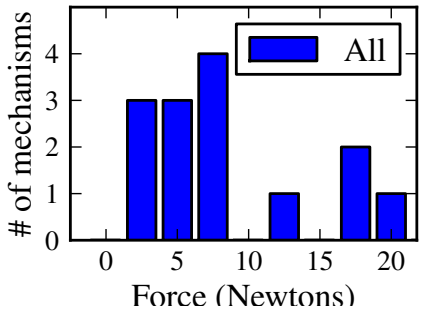
Figure 79: *Histograms of kinematic properties of drawers from our hand-measured kinematic dataset of 152 doors.*

in the forces for the remaining trials indicating that the data are dominated by configuration dependent forces, and thus agrees with our quasi-static assumptions. Furthermore, average velocity is a reasonable way to model the variation across trials, since all the doors and drawers were closed and at rest at the start of each trial, and the mechanism configuration increased monotonically during all trials.

For the data we use in our analysis, the angular velocities for door opening had a mean (standard deviation) of 19.6°/s (8.2°/s). The mean (standard deviation) of the average linear velocities for drawer opening was 0.14m/s (0.068m/s). Moreover, the structure of the forces relates to the configuration-dependent mechanics, rather than other aspects of the trajectory. As shown in fig. 73 on page 100 the forces when opening the cabinet were dominated by the non-linear spring that holds the cabinet closed and has a peak force at around 4°of opening. Likewise, when opening the drawer the forces were dominated by the effects of the telescoping rail.



(a) A histogram, colored by category, of the minimum forces to initiate motion while opening 31 doors. "Appliances" include refrigerators, freezers, and a microwave.



(b) A histogram of the minimum forces that were necessary to start opening the 14 different drawers in our dataset.

Figure 80: *Histograms of the minimum tangential forces that were necessary to initiate motion of 31 different rotary mechanisms and 14 drawers in our dataset.*

Figure 80a summarizes the minimum force that was required to start the motion of a door or drawer. We compute this as

$$f_{motion} = \min_{i \epsilon \{1 \ldots N_i\}} \max_{q \epsilon [0, q_0)} f_{tan}^{q,i}, \tag{51}$$

where q is the mechanism configuration, N_i is the number of trials for the i^{th} mechanism, and $f_{tan}^{q,i}$ is the tangential force at q for trial i. q_0 is a threshold that we used to determine when the motion of the mechanism began. We set it to 1° for rotary joints and 1cm for drawers. Figure 80a shows that there is a clear separation in the force required to initiate the motion of cabinet doors, freezers and refrigerators, and spring loaded doors in our dataset. We found that drawers tend to require less force to operate than rotary mechanisms. The maximum force required to start opening any drawer in our dataset was around 20N.

6.4.3 Principle Components Analysis

We now present trends in the data for rotary mechanisms revealed using Principle Components Analysis (PCA). We first convert the force profiles into a vector v^i for every trial i as

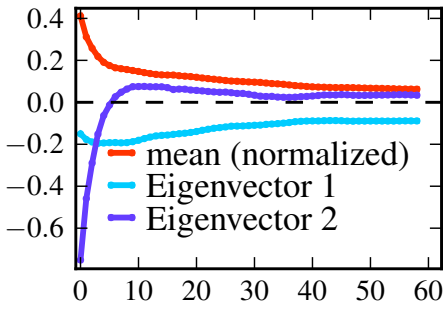
$$v^i = \begin{bmatrix} v_1^i & v_2^i & \ldots & v_{50}^i \end{bmatrix}^T, \tag{52}$$

where

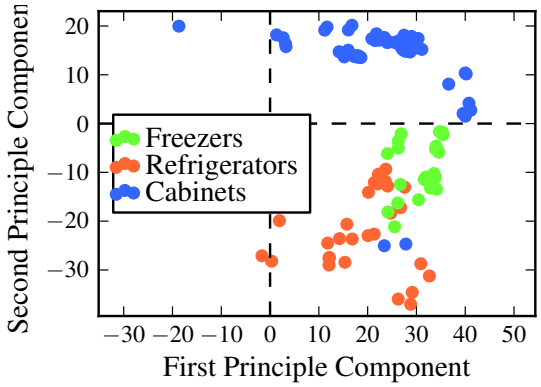
$$v_j^i = \max_{q \, \epsilon \, [j-1, j)} f_{tan}^{q,i}, \tag{53}$$

and q is the angle of the mechanism in degrees. We selected a maximum angle of 50° so that we could use data from almost all the force profiles. We then compute

$$V = \begin{bmatrix} v^1 & v^2 & \ldots & v^M \end{bmatrix}, \tag{54}$$



(a) The mean vector and the first two principal components of the force profiles.



(b) Scatter plot showing the projection of all the force profiles onto the first two principal components.

Figure 81: *Trends in the data using Principle Components Analysis.*

which is the matrix of all trials for all mechanisms combined, a total of M trials. In our dataset, we have a total of 146 trials on 29 rotary mechanisms.

Figure 81 shows the first two principal components and a scatter plot of these 146 trials projected onto the 2D subspace. Points are colored by mechanism type. The first two principal components together account for 91.8% of the data's variance (83% and 8.8%) and 8 principle components represent 99% of the variance over the 146 force profiles for 29 doors.

The scatter plot of fig. 81 shows that even after projecting to two dimensions, there is some separation between refrigerators, freezers and cabinets. These clusters appear to reflect common attributes of the mechanisms that keep the doors shut, see section 6.4.4 on page 110 and fig. 73 on page 100. The blue scatter points in the refrigerator cluster are from the only microwave in our dataset. Qualitatively, the microwave is similar to refrigerators as the forces that keep it closed decay rapidly as the mechanism is opened.

6.4.4 Illustrative Examples

In this section, we present examples from our dataset that illustrate that the non-linear behavior of doors and drawers is consistent across multiple trials for the same mechanism. Figure 73 on page 100 shows f_{tan} as a function of the mechanism configuration for all trials of four different mechanisms in our dataset. We can get some intuition about the shapes of the force profiles by looking at the mechanisms that keep them shut.

Refrigerators require a large initial force to open them due to the suction from the low pressure inside the refrigerator. However, once opened the doors swing easily due to bearings on the door. The force profile thus has a high peak when the door is closed and is very small after that.

Figure 82: *Three different types of drawers that we encountered – drawers with telescoping rails, rollers, and drawers with nothing preventing them from falling down as they are opened.*

Spring loaded doors require a large force to open. However, the tangential force required to open this spring loaded door is non-linear and decreases as the door is opened. Almost all cabinet doors also exhibit a behavior similar to the spring loaded doors. Most cabinets have a spring-like mechanism in their hinge that ensures that the door remains closed.

We observed that many drawers have rollers or telescoping rails. Telescoping rails can introduce a change in the amount of force required to open the drawers as different sections of the telescoping rail can have different mechanical properties, see fig. 73 on page 100. Finally, we also encountered some drawers that have neither rollers nor rails. These drawers tend to fall down as they are pulled out and might require more complex control methods compared to the other types of drawers. Figure 82 shows images of the three different types of drawers that we encountered.

6.5 Implications for Robot Design

By answering questions such as "How high are the handles of most doors and drawers?" and "What forces are necessary to open most doors and drawers?", our approach can help support the rational design of assistive robots. In this section, we discuss how our data could be used to inform design decisions for assistive mobile manipulators.

6.5.1 Related Approaches to Robot Design

Current approaches to the design of mobile manipulators for human environments fall into two main categories.

The first is to design robots by emulating aspects of human form and function [108]. The majority of human environments have been built to enable able-bodied humans to perform everyday tasks. If a robot were to perfectly emulate the capabilities of an average adult human, it would be capable of performing IADLs and providing assistance. The main problem with this approach is that humans represent an exceedingly high-bar for robot design. This is coupled with a lack of clear

guidelines on how to simplify the human model, while producing an effective system.

Second, many designers attempt to design robots by understanding the tasks the robot would perform. The main advantage of this approach is that it opens up the possibility for robot designs that differ dramatically from humans. For example, this could result in designs for robots with limited capabilities that are achievable in the near term, and designs for robots with super-human capabilities in the long-term.

Designers following this approach use qualitative assessments of the task constraints [48, 93, 225] or government and industry standards to guide their designs [215]. In the US, for example, public and commercial spaces are legally regulated by the Americans with Disabilities Act to conform to accessibility guidelines (ADAAG) [6]. Among other specifications, these guidelines dictate that doorways need to be at least 36" wide and doors need to be equipped with lever handles. There are also industry standard dimensions for some home features, such as cabinets, drawers, and countertops. However, in spite of these efforts at standardization, significant variation remains in the real world. For example, residential homes are exempt from the ADAAG and older buildings may be out of compliance. Additionally, guidelines and requirements can change over time, allow for non-trivial variation, and leave critical mechanical properties unspecified, such as the forces that operate devices.

The strength of our approach is that it has the potential to quantitatively assess the statistics of real-world mechanics, instead of following intuition, heuristics, or written design standards. Challenges to our approach include factoring the mechanics specific to IADLs from the mechanics of the capturing process. Moreover, for this approach to be well-justified, the real-world statistics must be sufficiently complex to negate the possibility of simple specifications, and sufficiently structured for the captured data to be informative. Recent research by Aaron Dollar et al. has pursued a similar approach in the context of objects [136].

6.5.2 Influencing Robot Design Through Captured Kinematics

We now discuss ways in which the captured kinematic data could be used to inform the design of an assistive robot that opens doors and drawers. Kinematic data provides necessary conditions that the robot must meet to quasi-statically operate doors and drawers by interacting with the handle. If we assume that the robot's end effector makes contact with the handle, then the end effector must be able to traverse the trajectory the handle follows when opening the door or drawer to a desired configuration.

For a conventional mobile manipulator, this immediately implies constraints on the workspace

of the arm, and the interplay between the motion of the arm and the motion of the mobile base. For example, if the end effector can not reach a range of heights, the robot would be unable to operate doors and drawers at those heights, see figs. 78a and 79a on pages 107–108. Likewise, the distribution for different radii of doors (fig. 78b on page 107), and the distance drawers can be pulled out (fig. 79b on page 108) imply particular trajectories that the end effector would need to traverse, either by movement of the arm, movement of the mobile base, or a combination of the two.

One of the main opportunities provided by this data, is the potential to rationally tradeoff features such as the arm's workspace, the agility of the mobile base, the cost, and the robot's success at operating various doors and drawers. One extreme design would use long arms that can traverse these trajectories without moving the base. For this design, a simple base might be sufficient, but the arms would probably be more complex, since a longer reach implies higher joint torques and greater positional errors due to joint angle errors. Likewise, a design that combines very short arms with an agile base might be feasible. Given the statistics, these tradeoffs could be considered relative to the desired performance over various doors and drawers. For example, an assistive robot that only operates low drawers and doors of moderate size might improve quality of life sufficiently to be desirable, especially if the cost were substantially reduced.

Kinematic data could also be used to improve software design. For example, prior knowledge can improve estimation of the kinematics of mechanisms [207]. Robots might also use this information to infer the type of door or drawer based on an observed trajectory. For example, a large radius with a handle one meter above the ground is unlikely to be a kitchen cabinet.

6.5.3 Influencing Robot Design Through Captured Forces

We now discuss how the captured forces could be used to inform the design of an assistive robot that opens doors and drawers. If we assume that we have accurately captured the quasi-static forces associated with opening doors and drawers, then we can define necessary conditions for an assistive robot to open these same mechanisms. In other words, the data defines a lower bound on the required forces. Any increase in velocity or acceleration would result in an increase in the force. Likewise, our quasi-static assumption implies that reductions in velocities or accelerations would not lower the forces significantly.

Consequently, the captured forces can be used to provide guidelines for a robot's power. For the given end effector velocities, a robot must be able to generate the necessary forces at the end effector. The implications of these constraints are particularly evident with respect to different classes of doors and drawers. For example, from fig. 80a on page 109, springloaded doors, refrigerators and

freezers require much greater force to open than cabinets, so a robot that only opens cabinets would have much lower power requirements. Likewise, we found that drawers require less force to operate than rotary mechanisms.

Researchers have demonstrated that dynamically stable mobile manipulators can apply more force when pushing by leaning into the action. For example, in [212] a dynamically stable and statically stable version of a small-scale mobile manipulator attempted to push a drawer closed. The statically stable version failed, while the dynamically stable version succeeded. The authors used this result to argue for the superiority of dynamically stable mobile manipulators, but they neglected other engineering challenges associated with dynamic stability, such as increased complexity and safety concerns. By capturing the forces required to close drawers, we could estimate the value of this additional force. For example, if the greater forces provided by dynamic stability were unnecessary for a particular assistive robot design, then a statically stable base might be more appropriate.

Finally, force data can be used to improve the robot's software. Although the mechanics of operating real doors and drawers is non-linear, the structure exhibits consistency within classes across houses, such as kitchen cabinets, freezers, and refrigerators. This suggests that a robot could use similar data to better detect anomalous conditions, such as collisions, locks, or obstructions. Likewise, the robot might use this data to decide how much force to apply when opening a door or drawer for the first time. In addition, robots might be able to haptically infer characteristics of doors and drawers. For example, the forces associated with opening a refrigerator are highly distinctive.

In chapter 7, we present methods by which the force data of human operators opening doors that we captured in this chapter can be used by robots to detect anomalous conditions while opening doors, and haptically recognize the mechanism that it opened.

6.6 Future Work

There are many opportunities for additional research in this area. Further analysis and better estimation of the hook's velocity and acceleration could help validate our quasi-static assumptions. Faster operation of doors and drawers might be preferred by users of assistive robots, which would require methods to capture and characterize highly-dynamic mechanics. For example, we have observed humans "throwing and catching" doors and drawers by initially applying an impulsive force and then decelerating the door or drawer at a desired configuration. Variation in mechanisms over time is another complexity that could be addressed. For instance, the mass of a drawer changes based on its contents, and some doors have objects that hang from them.

6.7 Conclusion

Using a custom motion/force capture system, we have shown that operation of seemingly simple mechanisms like doors and drawers involves significant complexities, including large initial forces and non-linear forces. We have also shown that relevant kinematic parameters, such as handle height and the distance drawers can be opened, exhibit large variation. We have demonstrated that in spite of these complexities, the data exhibit structure that can be used to inform the design of assistive robots.

Based on our results, we are optimistic that everyday mechanics associated with other aspects of IADLs can be captured, characterized, and used to improve assistive robots. We expect that collecting large scale datasets will become easier as robots are deployed in human environments and interconnected through the internet.

CHAPTER VII

IMPROVING ROBOT MANIPULATION WITH DATA-DRIVEN OBJECT-CENTRIC MODELS OF EVERYDAY FORCES

Robots currently lack common sense about the forces involved in everyday manipulation tasks. In chapter 6, we described our progress towards capturing and characterizing the forces required to open doors and drawers. In this chapter, we present data-driven methods to inform robots about the forces that they are likely to encounter when performing specific tasks. In the context of door opening, we demonstrate that data-driven models can be used to haptically recognize specific doors, haptically recognize classes of door (e.g., refrigerator vs. kitchen cabinet), and better detect haptic events (e.g., a locked door or collision), even when opening a specific door for the first time. We also show that two distinct robots can use forces captured from people opening doors to better detect haptic events. This illustrates the potential for databases of task-specific forces to be captured by people and robots to inform robot manipulation. The work presented in this chapter is currently under review [89].

7.1 Overview of Our Approach

Vast quantities of captured real-world auditory and visual data are available to humans and machines, but captured real-world haptic data is rare. Capturing the forces associated with everyday manipulation tasks could benefit robots by enabling them to better interact with the physical world.

In this chapter, we present data-driven methods to inform robots about the forces that they are likely to encounter when performing specific tasks. We show that data-driven models can be used to haptically recognize an instance of a mechanism, haptically recognize a mechanism class, and haptically detect anomalous events. We show that the forces captured while distinct robots and people perform a task can be comparable, and that distinct robots can use forces captured from people to better detect haptic events. This demonstrates the feasibility of capturing datasets of task-specific forces to inform robots.

We focus on the example task of pulling open a door. Door opening is an important task for mobile manipulators, and a number of researchers have demonstrated robots with autonomous door opening capabilities. However, enabling autonomous robots to robustly open doors in the real world

remains a challenge. Door opening robots have lacked compelling ways to deal with many common situations, such as a door that is locked, blocked, damaged, or colliding with something. For example, a robot needs to decide how much force to apply to a locked door before it gives up. If the robot gives up too early, it will fail to open unlocked doors that require high forces, such as refrigerators. If the robot gives up too late, it will waste time and apply unnecessary force that could cause damage or be unsafe. We provide evidence that models of real-world forces can be used by robots to better handle these common situations.

The models we present have the following three important characteristics:

1. ***Task-specific:*** Each model is specific to a narrowly defined manipulation task. This is intended to reduce the complexity of the model and the data requirements.

   For this paper, we defined the task to be smoothly and slowly pulling open a door with contact restricted to the handle.

2. ***Data-driven:*** The models directly use forces, points of application of the forces, and kinematics captured during real-world performance of the task in a variety of representative circumstances. This is intended to capture the natural variation that a robot will encounter.

   For this work, we used data captured while opening 28 doors in 6 homes and one office in Atlanta, GA, USA, which we described in chapter 6 on page 99 and [96].

3. ***Object-centric:*** Each model relates the relevant state of the manipulated object to the relevant forces applied to the object. This is intended to make the models independent of the robot or human manipulating the object, and thus enable the models to be useful for distinct robots and methods of manipulation. For example, it should not matter how the robot applies the forces, whether with its left hand, its right hand, or its elbow.

   In this work, the models are quasi-static. They relate the opening angle of the door to the component of the force applied to the handle that is tangential to the motion of the door.

7.1.1 Organization of this Chapter

The rest of the chapter is organized as follows. In section 7.2 on the next page, we discuss related work and contrast it with our approach. In sections 7.3 and 7.4 on pages 121–123, we describe our methods of data capture, a quasi-static model for doors, and the object-centric representation that we use. Next, in section 7.5 on page 127, we show that a standard supervised learning classifier can recognize the class of a door (e.g., refrigerator or kitchen cabinet) and the specific instance of a door

based on the forces during opening. In section 7.6 on page 130, we present a probabilistic model of the forces encountered while successfully opening a mechanism.

Using this probabilistic model, we present a method for detecting haptic events, such as locked doors or collisions, in section 7.7 on page 131. We compare this method with two baseline methods for haptic event detection and show that a robot can use previously captured haptic data to detect locked doors and collisions more quickly and with lower applied forces. In section 7.8 on page 135, we report results of haptic event detection from trials on two distinct robots and discuss the implications of online state estimation on the performance of haptic event detection. Finally, we discuss limitations of our work and directions for future research in section 7.9 on page 140 and conclude in section 7.10 on page 142.

7.2 Related Work

7.2.1 Capturing Haptic Stimuli

Although vast quantities of visual and auditory stimuli can be easily accessed on the web, very little haptic stimuli can be found. Researchers have looked at capturing haptic interactions to synthesize realistic haptic sensations for human users, for example [149, 51, 129, 17, 217, 178]. To date, however, this body of work has emphasized high-fidelity models of objects to convey realistic haptic sensations to people, rather than capturing haptic datasets to inform robots.

Rehabilitation research has collected datasets of pinch forces during activities of daily living (ADLs) with the goal of using them as a quantitative measure of recovery from hand surgery and to inform prosthetic hand design (e.g, [197, 224]).

As described in chapter 6, we captured the forces applied to door handles and the trajectories of the handles while pulling open doors with the motivation of informing the design of robot hardware and software. Other research has collected datasets of forces during some everyday activities [174] and physical properties of objects [136] with similar motivation. However, these datasets have not yet been used to inform robot manipulation.

7.2.2 Haptic Recognition and Anomaly Detection

In contrast to our use of data-driven object-centric models, previous research on haptic recognition and anomaly detection for robot manipulation has often used data-driven models that are robot-centric or models of the dynamics of the robot arm.

7.2.2.1 Data-driven and Robot-centric Models

Researchers have used data-driven robot-centric models of haptic data in the form of tactile sensor arrays, joint torques, and joint angles to haptically recognize objects grasped by a robot hand (e.g., [68, 101, 210]). [196] demonstrated that haptic data in the form of joint torques associated with specific behaviors for the entire arm can also be used to recognize objects.

Other research has used robot-centric models to detect anomalous conditions during a manipulation task. For example, [176] has demonstrated that force data captured during an assembly operation can be used to predict failure in future trials. [152] has shown that a database of joint angles, joint torques, tactile sensor information, and accelerometer data can be used to predict failure as a robot flips a box using chopsticks, or plays a pool stroke. [208] used deviations from a data-driven model of joint torques during free-space swiping motions to detect conditions when a magnetic card gets stuck as a robot swiped it through a card reader.

For these methods, the state of the robot is intertwined with the haptic representations. For example, these methods often use ego-centric sensor data parameterized by time or the robot's state, such as joint angles. As a result, there is no direct way to combine captured data from distinct robots and people. Nor is there a direct way for distinct robots to share what they have learned.

[191] has presented methods for object identification with bag-of-features models using haptic data in the form of readings from tactile sensor arrays on the robot's parallel jaw gripper, and the width and height at which the robot grasps the object. Although [191] presented results from data collected by a single robot, these models are data-driven and object-centric and different robots with similar sensing capabilities may be able to share these haptic data.

7.2.2.2 Anomaly Detection Using Joint Torque Sensors and Arm Dynamics

Previous research has used deviations from an expected torque, predicted using a model of the dynamics of the robot arm, to detect anomalous conditions (e.g, [45, 40, 71, 72]).

Determining an accurate model of the arm dynamics can be challenging. Additionally, these approaches often detect anomalous conditions in free-space motions. Estimating expected torques in situations where the robot makes contact with its environment is more complex (e.g, [143]). For example, while pulling open doors, forces along the radial direction (and hence the torques at the joints) can change without triggering an anomalous condition as discussed in section 7.3.1 on page 121. In this case, the representation of acceptable torques would have to be more complex. Likewise, it would be desirable for robots to be able to operate doors for which careful mechanical modeling has not been performed.

7.2.3 Robotic Door Opening

Doors are ubiquitous in indoor human environments and form an important sub-class of mechanical systems. For robots to autonomously move between rooms, they will often need to open doors. People who have difficulty opening doors for themselves might also benefit from robotic door opening. Moreover, many objects within human environments are stored behind doors, such as in refrigerators and cabinets.

Recently, researchers have developed a number of robotic systems to operate doors between rooms (e.g, [114, 91, 215, 34, 103, 116]), and open cabinets, drawers, and appliances (e.g, [94, 221, 44, 180, 19]). These efforts have focused on controllers and planners that enable a robot to open doors. They have not addressed haptically detecting collisions between the door and the environment, or haptically identifying the mechanism identity or class.

7.2.4 Relation to Our Previous Research

We have previously developed methods that enable a robot to autonomously open doors and drawers without prior knowledge of the mechanism kinematics, described in chapter 5 on page 63. We use that system in this chapter to enable the robots Cody and a PR2 to open doors, see fig. 87 on page 126.

Additionally. in collaboration with Sturm, Stachniss, and Burgard, we have shown that a robot can use a database of kinematic trajectories of mechanism handles to increase the online prediction accuracy of the kinematic state of a mechanism that it is currently opening [205]. We do not use our research from [205] in this chapter. However, it is complementary to the current chapter as it looks at kinematic data, and this work investigates haptic data.

Lastly, as described in chapter 6 on page 99, we built a force and motion capture system and used it to capture the forces and kinematic trajectories of the handle of doors and drawers as human operators opened these mechanisms. We showed that the forces can be represented in a way that is intrinsic to the mechanism, and discussed the implications of this database of haptic interactions for assistive robot design. In this chapter, we demonstrate how robots can improve their manipulation performance if they have access to a database of haptic interactions by using the database that we created in chapter 6.

7.3 Capturing Haptic Interactions

In this section we describe how we capture the haptic interactions for humans and two robots. For the task of pulling open doors by the handle, we use the term haptic interaction to refer to the relation between the component of the force applied to the handle that is tangential to its trajectory and the angle through which the door has been opened.

7.3.1 Quasi-static Model of Doors

Our methods for haptic identification and haptic event detection rely on modeling the relationship between the relevant forces, f, applied to a mechanism, m, and the mechanism's state, x. As such, we need to make measurements to obtain estimates of the relevant applied forces, $\hat{f}$, and the relevant state of the mechanism, $\hat{x}$.

In this work, our kinematic model of doors is a single degree of freedom (DoF) rotary joint whose axis of rotation is parallel to gravity, as shown in fig. 83. We have found that at relatively low opening speeds, the configuration dependent forces dominate the haptic interactions [96]. So, we assume that the relevant state consists solely of the opening angle of the door.

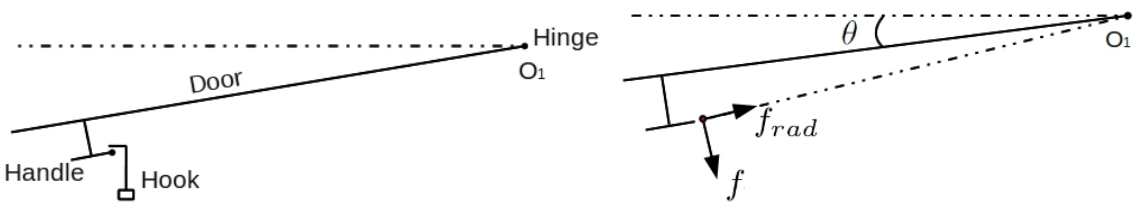


Figure 83: ***Left:*** *Diagram showing the top view of a door as it is opened using a hook.* ***Right:*** *The relevant force, f, is the component of the total force between the hook and the handle that is tangential to the trajectory of the door handle. The radial force, f_{rad}, will result in constraint forces at the hinge. The relevant mechanism state is the opening angle, x.*

Additionally, we assume that the only relevant force consists of the component of the total force applied to the handle that is tangential to the handle's trajectory. This is consistent with the rotary joint model of doors where the tangential component of the applied force will open the mechanism, while other components will result in constraint forces at the hinges. For more details of this quasi-static model of doors, we refer the reader to section 6.3 on page 104.

7.3.2 Estimating the Relevant Applied Force and the Relevant Mechanism State

In our experiments, we use a rigid 3D printed hook instrumented with a six-axis force-torque sensor (ATI Nano25 with a calibration of SI-125-3), shown in fig. 84, to measure the forces applied to the door.

Figure 84: *The hook with a force-torque sensor at its base that we used for the three data capture systems.* ***Left:*** *Handheld hook used by human subjects.* ***Middle:*** *The hook end effector mounted on Cody.* ***Right:*** *The hook with an adaptor that the PR2 can grasp with its gripper.*

To estimate the mechanism state, we first measured the trajectory of the door handle as a human operator (section 7.3.3 on the next page) or a robot (section 7.3.4 on the following page) opened the door, using a motion capture system or forward kinematics respectively. We then fit a circle to this trajectory to estimate the radius and the location of the axis of rotation of the door. This enabled us to estimate the angle of the door for each point of the trajectory.

We also used the estimated angle of the door to compute the component of the force measured by the force-torque sensor that was tangential to the trajectory of the handle. We detail our method for estimating the door angle and tangential force in sections 6.2 and 6.3 on pages 103–104.

7.3.3 Capturing Forces Applied by Humans

In chapter 6, we captured the forces and kinematic trajectories as human operators opened 28 doors in 6 homes and one office in Atlanta, GA, USA. To do this we built a custom force and motion capture system that consisted of a hook end effector instrumented with a force-torque sensor and checkerboard patterns to track the trajectory of the mechanism and the hook shown in fig. 74 on page 101. This enabled us to generate a database of estimates of the tangential force applied to the handle ($\hat{f}$) as a function of the estimated opening angles of the door ($\hat{x}$). We use this database to present results on haptic identification in section 7.5 on page 127, and haptic event detection in section 7.7 on page 131.

7.3.4 Capturing Forces Applied by Robots

We use a feedback controller that we developed in [94] to enable two robots, Cody and a PR2, to autonomously open a door without prior knowledge of the kinematics. We mounted the same instrumented hook end effector on both robots, as shown in fig. 84 on this page. Both robots use the same force feedback controller to autonomously open doors, but have different low-level control. We use joint space impedance control on Cody [94] and a Cartesian space stiffness controller on the PR2 [65].

While each robot was opening a door, we recorded the trajectory of the tip of the hook (using joint encoders and forward kinematics), as well as the force measured by the force-torque sensor. We then used the method described in section 7.3.2 on the previous page to estimate tangential force applied to the handle, $\hat{f}$, and the angle of the door, $\hat{x}$. We use haptic data from trials with the robots along with the database of haptic interactions of humans to report results of haptic event detection in section 7.8 on page 135.

7.4 Common Database of Haptic Interactions

We now describe our representation of haptic interactions and illustrate how humans and two robots can contribute to a common database of haptic interactions.

7.4.1 Representing a Haptic Interaction

As described in section 7.3 on page 121, we record estimates of the applied tangential force and the mechanism's state as a sequence of N tuples $\{\{\hat{f}_1, \hat{x}_1\}, \{\hat{f}_2, \hat{x}_2\}, ...\{\hat{f}_N, \hat{x}_N\}\}$. This sequence of tuples is a raw haptic interaction.

Figure 85 on the following page shows examples of raw haptic interactions captured when people opened the pictured mechanisms at low speeds. For opening doors, the handle defines the location at which the instrumented hook applies forces to the mechanism. More generally, the haptic interaction would also have to include the point of application of the force relative to the object that is being manipulated.

We further process this raw haptic interaction into a more compact and uniform representation. We first quantize the opening angle of the door into 1° intervals. We then represent each haptic interaction by a fixed length vector, where each element in the vector is set to the mean tangential force encountered in the corresponding 1° interval, or set to NaN if the interval was not encountered. In this chapter, we will refer to this vector as the haptic interaction vector.

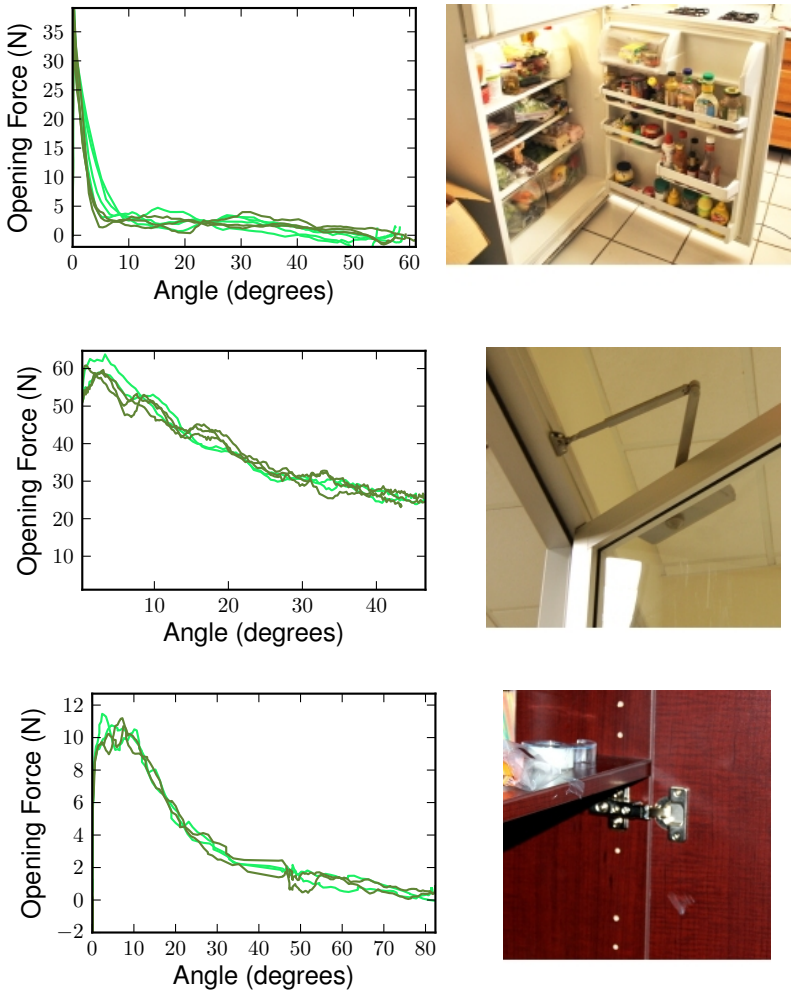


Figure 85: *Forces recorded while opening three doors. Left plots show forces tangential, f_{tan}, to the motion of the handle as a function of the device's configuration. Lighter green indicates trials with higher average velocity. Pictures on the right highlight a key mechanical element of each mechanism.* ***Top:*** *refrigerator, 6 recordings, avg. velocities of 17.8°/s to 26.8°/s. High initial force due to low pressure interior.* ***Middle:*** *springloaded door, 5 recordings, avg. velocities of 6.5°/s to 13.5°/s. Large forces throughout movement due to linkage at top.* ***Bottom:*** *kitchen cabinet, 9 recordings, avg. velocities of 7.4°/s to 16.3°/s. Non-linear spring keeps it closed with max force at about 4°.* This figure uses the same data as fig. 73 on page 100.

7.4.2 Sharing Haptic Data

In this section we illustrate that object-centric haptic interactions can be shared among people and distinct types of robots. We show that this haptic data can be insensitive to some forms of task variation, such as the robot's position in the environment. We also show that different people and robots can have similar haptic interactions when manipulating the same mechanism.

First, fig. 86 on the next page illustrates the small variation in haptic interaction resulting from

changes in the PR2's position relative to the handle of a cabinet. Second, fig. 87 on page 126 shows the mean and standard deviation (over multiple trials) of data from four sets of trials in which two humans and both the robots opened the same mechanism. The variation in the tangential force due to the operator is relatively small, showing that this component of the forces associated with the manipulation task is intrinsic to the mechanism.

This observation, combined with our results throughout this paper, demonstrates that robots and humans can share haptic data through a common database of haptic interactions while pulling open doors.

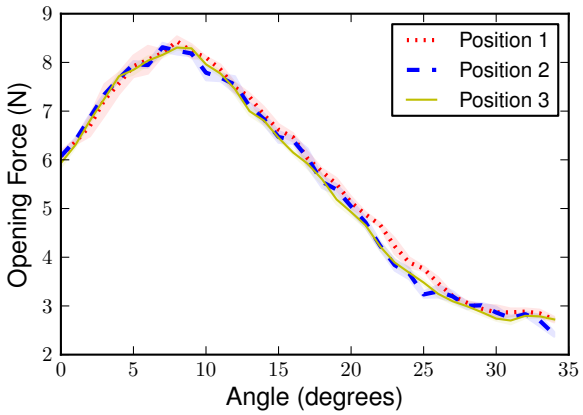
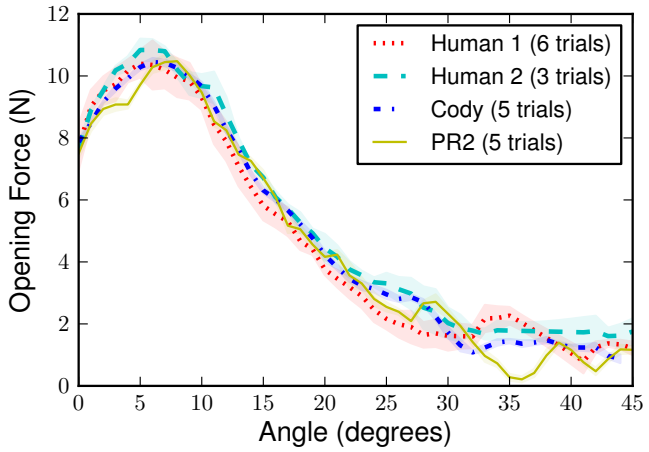


Figure 86: *Data from the PR2 opening the same mechanism five times each from three different positions relative to the mechanism handle. The mean and standard deviation of the measured tangential force, $\hat{f}$, as a function of mechanism configuration.*

(a) Human

(b) Cody

(c) PR2



(d) Measured tangential force

Figure 87: *Haptic data can be shared. When opening the same mechanism using an instrumented tool* ***(a)-(c)****, two humans and two robots have very similar haptic interactions in spite of their distinct bodies and control.* ***(d):*** *The mean and standard deviation of the tangential force as a function of mechanism configuration for multiple trials.*

7.5 Haptic Identification

In this section, we show that standard supervised learning classifiers applied to the captured haptic data can be used to recognize the class of a door (e.g., refrigerator or kitchen cabinet), as well as the specific door. We used a data set of 148 haptic interaction vectors from human subjects opening 28 different doors.

7.5.1 Dimensionality Reduction

To reduce the influence of noise and overfitting, we first computed a low dimensional representation of the haptic interaction vectors with Principal Component Analysis (PCA). As described in section 6.4.3 on page 109, the first two principal components together account for 91.8% of the data's variance (83% and 8.8%) and 8 principle components represent 99% of the variance over the 146 force profiles for 29 doors. fig. 81 on page 110 shows the first two principal components and a scatter plot of the force profiles projected onto the 2D subspace.

7.5.2 Recognizing a Specific Mechanism

We now present results on haptically recognizing a specific door after opening it. We assume that the database of haptic interactions contains trials from opening that specific mechanism.

Figure 88 on the next page shows the leave-one-out cross-validation error for a k-nearest neighbor classifier (k=1 and k=3) and a support vector machine on our data set for subspaces of different dimensionality. The cross-validation error with the kNN classifier (k=1) and the SVM were similar. The error was nearly constant for subspaces of dimensionality ≥ 5. Figure 89 on the following page shows the confusion matrix for the kNN classifier after projecting the data onto the first five principal components. With these 28 specific mechanisms, the cross-validation success rate for identifying the mechanism was 91.8%.

Confusion occurred between mechanisms for which the tangential forces are similar. Scaling this database up to a very large size might reveal further subclasses of mechanism, such as doors made by various manufacturers.

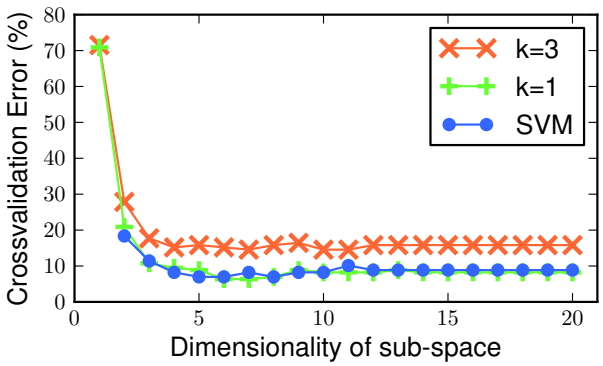


Figure 88: *Crossvalidation error as a function of the dimensionality of the subspace for three classifiers.*

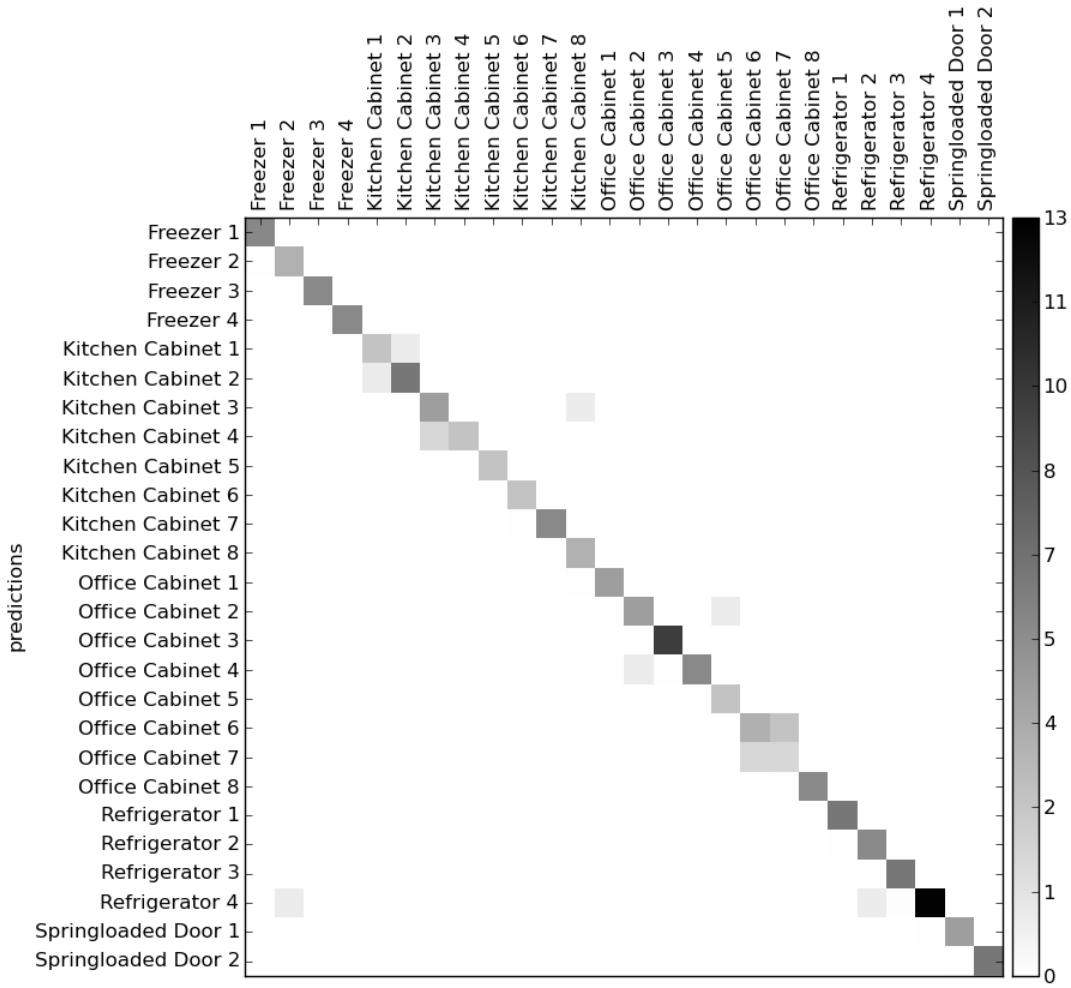


Figure 89: *Confusion matrix from leave-one-out cross-validation with a kNN classifier (k=1), after dimensionality reduction to five using PCA. The goal is to identify a mechanism (see section 7.5.2). The cross-validation error was* 8.2%.

7.5.3 Recognizing the Mechanism Class

In this section we look at the problem of identifying the class of a mechanism after a human or a robot opens it. We assume that our database of haptic interactions includes mechanisms in the same class, but does not include the specific mechanism.

We assigned the class labels of "freezer", "refrigerator", "kitchen cabinet", "office cabinet", and "spring loaded door" to each trial in our database. We did not consider microwaves in this evaluation, since we only have data from a single microwave.

We used a kNN classifier (k=1) and a 5 dimensional linear subspace (PCA) for class recognition. For a selected mechanism, we generated a training set by removing all the vectors from that mechanism from our dataset. We did this to simulate opening the mechanism for the first time. We then tested the kNN classifier for each of the vectors from the selected mechanism, and repeated this procedure for all 28 mechanisms.

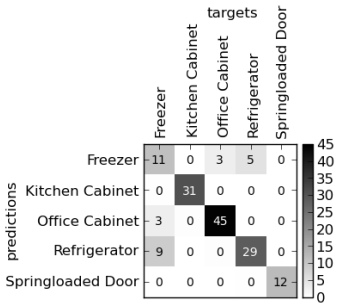


Figure 90: *Confusion matrix from cross-validation with a kNN classifier (k=1), after dimensionality reduction to five using PCA. The goal is to identify the mechanism class. The cross-validation error was* 13.6%.

Figure 90 shows the confusion matrix and the cross-validation error for our data set. The cross-validation success rate for identifying the semantic class of the 28 mechanisms, given that the specific mechanism had not been encountered before, was 86.4%. Most of the classification errors were between the semantic classes refrigerators and freezers. Both these classes have similar tangential forces as a function of the configuration, with the major difference being the initial force required to open them.

7.6 A Probabilistic Model of Applied Force

In this section, we present a probabilistic model of the relevant force, f, applied while successfully operating a mechanism, m, conditioned on the relevant mechanism state, x, its class, C, and any previous data from the specific mechanism, D_x. D_x is a vector of any forces previously measured at state x while operating mechanism m.

We model the relevant applied force at a particular mechanism state x as being normally distributed and conditionally independent of other forces, given x, C, and D_x. So,

$$P(f|x, C, D_x) \quad = \quad \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(f-\mu)^2}{2\sigma^2}}, \tag{55}$$

where μ is the mean and σ^2 is the variance of our Gaussian model for the relevant force at mechanism state x. Our goal is to estimate μ and σ^2 given x, C, and D_x.

7.6.1 Operating a Mechanism for the First Time

Consider the case when a robot operates a mechanism for the first time. We assume that the robot knows the class S to which the mechanism belongs. For example, using vision and knowledge of the room type, the robot might know that the door is a kitchen cabinet. The robot also has access to a database of haptic interaction vectors from mechanisms that are members of class S.

In this situation, the robot knows the mechanism's state x and the mechanism's semantic class S, but D_x is a zero-dimensional vector with no information, since the robot has not previously operated the mechanism. We use the haptic data from mechanisms of semantic class S to estimate μ and σ^2 with a weighted sample mean and weighted sample variance, $\hat{\mu}$ and $\hat{\sigma}^2$ [26]. So,

$$\hat{\mu} = \frac{\sum_{m \epsilon S} \left(w_m \sum_i \hat{f}^i_{m,x} \right)}{\sum_{m \epsilon S} \sum_i w_m} \tag{56}$$

$$\hat{\sigma}^2 = \frac{\sum_{m \epsilon S} \left(w_m \sum_i \left(\hat{f}^i_{m,x} - \hat{\mu} \right)^2 \right)}{\sum_{m \epsilon S} \sum_i w_m} \tag{57}$$

where $\hat{f}^i_{m,x}$ represents the element of the i^{th} haptic interaction vector for mechanism m corresponding with mechanism state x. $m \epsilon S$ represents selecting a mechanism from all mechanisms in semantic class S. The weight for mechanism m is

$$w_m = \frac{1}{\text{\# of haptic interaction vectors for } m}. \tag{58}$$

7.6.2 Operating a Mechanism for the n^{th} Time

Now, consider the case when a robot operates a mechanism that it has previously operated. In this situation, the robot knows the mechanism's state x, the mechanism's class S, and D_x, which is a vector with previous forces for this specific mechanism at state x. In this case, we make a maximum a posteriori (MAP) estimate [26] of μ and σ^2 with the following equation:

$$(\hat{\mu}, \hat{\sigma}^2) = \underset{(\mu,\sigma^2)}{\operatorname{argmin}} \{-\log P(\mu, \sigma^2 | x, S, D_x)\}. \tag{59}$$

We use Bayes' rule to obtain

$$P(\mu, \sigma^2 | x, S, D_x) = \frac{P(D_x | \mu, \sigma^2, x, S) P(\mu, \sigma^2 | x, S)}{P(D_x | x, S)}. \tag{60}$$

Assuming that the forces from the previous $n-1$ operations of the mechanism were independently drawn from $\mathcal{N}(\mu, \sigma^2)$, then

$$P(D_x | \mu, \sigma^2, x, S) = \prod_i \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(D_x^i - \mu)^2}{2\sigma^2}}. \tag{61}$$

We model $P(\mu, \sigma^2 | x, C)$, the prior distribution over (μ, σ^2) given the state x and class C, as a normal distribution with mean $(\mu_\mu, \mu_{\sigma^2})$ and covariance matrix $diagonal(\sigma_\mu^2, \sigma_{\sigma^2}^2)$. We estimate these four parameters by first computing the sample mean $\mu_{x,m}$ and sample variance $\sigma_{x,m}^2$ of the measured tangential forces at state x over all mechanisms m in class C. We then compute $(\hat{\mu}_\mu, \hat{\mu}_{\sigma^2})$ by concatenating the sample means of $\mu_{x,m}$ and $\sigma_{x,m}^2$. Likewise, we compute $(\hat{\sigma}_\mu^2, \hat{\sigma}_{\sigma^2}^2)$ by concatenating the sample variance of $\mu_{x,m}$ and $\sigma_{x,m}^2$.

Equation (59) now simplifies to

$$(\hat{\mu}, \hat{\sigma^2}) = \underset{(\mu,\sigma^2)}{\operatorname{argmin}} \sum_i \left(\log \sigma + \left(\frac{D_x^i - \mu}{\sigma} \right)^2 \right) + \left(\frac{\mu - \hat{\mu}_\mu}{\hat{\sigma}_\mu} \right)^2 + \left(\frac{\sigma^2 - \hat{\mu}_{\sigma^2}}{\hat{\sigma}_{\sigma^2}} \right)^2. \tag{62}$$

We find approximate solutions for eq. (62) using the implementation of the BFGS optimization algorithm from SciPy [102] with seed estimates $\mu = \sum D_x^i/(n-1)$ and $\sigma^2 = \hat{\mu}_{\sigma^2}$.

7.7 Haptic Event Detection

We use our probabilistic model of forces applied to a mechanism, described in the previous section, to present a method for detecting anomalous forces. For comparison, we also present two baseline methods that do not use haptic data and use our probabilistic model to demonstrate the benefits of haptic data in detecting anomalous forces.

An example of the potential benefit of haptic data can be observed from fig. 91, which shows the maximum force required to open five different classes of rotary mechanisms by 10°, as computed

from human captured data. Figure 91 illustrates that knowledge of a mechanism's class can enable a robot to better select a maximum force to apply to a door before deciding that it is locked. This could vary from less than 10N for kitchen cabinets to 40-50N for refrigerators.

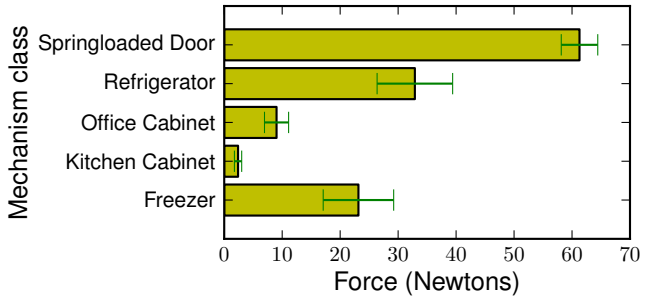


Figure 91: *Maximum force (mean and standard deviation) expected while opening mechanisms of five different classes of rotary mechanisms by an angle of 10°.*

In section 7.7.4 on page 133 we show that captured haptic interactions can be used to improve manipulation in two ways. First, by using captured haptic interactions, an anomalous force detector can reduce the increase in force from the onset of a collision until it detects the collision. Second, captured haptic interactions enable the detector to report collisions faster. The first improvement corresponds to an increase in the safety of the system by lowering the excess force applied to the door and the second corresponds to an improvement in the efficiency.

Cameras and other non-contact line-of-sight sensors are not well matched to many of these detection problems as the event naturally occurs as an anomalous force. For example, a door gently touching and deforming a curtain may not be an anomalous condition. Also, there may not be any clear visual cue for a locked door.

7.7.1 Three Methods for Detecting Anomalous Forces

We detect an anomaly if the force measured at the current state of the mechanism, x, exceeds a threshold force, i.e.

$$\hat{f}_x > f_x^{thresh}. \tag{63}$$

For this work, we do not investigate the potential for a lower bound on the force, although this could also be indicative of important haptic events.

We present three methods of determining f_x^{thresh}. The first method uses the probabilistic model from section 7.6 on page 130 to detect when forces are unlikely given the mechanism's state x, the mechanism's class C, and any previous operation of the mechanism D_x. For this method,

$$f_x^{thresh} = \hat{\mu} + n\hat{\sigma}, \tag{64}$$

where the parameter n trades off the false positive rate (percentage of times $\hat{f}$ is incorrectly reported as anomalous) with the sensitivity of the anomaly detection.

The other two detectors are baseline methods that use no prior information about the mechanism or its class. The first baseline detector sets f_x^{thresh} equal to a constant c. The second baseline detector sets $f_x^{thresh} = r \cdot \hat{f}_{initial}$, where r defines a fixed ratio of the initial opening force. We have used this method in our previous work described in chapter 5 on page 63. For these two detectors, c and r trade off the false positive rate with the sensitivity in an analogous way to n.

7.7.2 Performance Measures for Haptic Event Detection

We evaluated the performance of anomalous force detection methods using: 1) the increase in the magnitude of the force from the onset of the collision until its detection; 2) how much time passes between the onset of the collision and its detection; and 3) the false positive rate. Lower force implies less risk of damage to the robot, the environment, and nearby people. Faster detection implies that the robot can more efficiently respond to the haptic event, such as by trying a new strategy or stopping and asking for assistance. A lower false positive rate implies that the robot will be less likely to falsely detect an event and thereby unnecessarily change its approach or give up.

7.7.3 Modeling Locked Doors and Collisions with a Fixed Rigid Obstacle

Our database of haptic interactions consists of collision-free trials only. To compare the performance of these three anomaly detection methods using this database, we simulated locked doors, and collisions between the door and a fixed rigid obstacle.

We modeled these situations as a force that increases monotonically with time while the configuration of the mechanism remains constant. The actual rate of increase of the force over time will depend on the robot control algorithm. For example, for an impedance controlled robot, we would expect the force to increase at a rate that depends on the stiffness at the end effector.

This model allowed us to simulate collisions at any configuration of the mechanism. We could then compare the three anomaly detectors using our database of collision-free trials on the first performance measure, the increase in the magnitude of the force from the onset of a collision until its detection.

7.7.4 Evaluation of Collision Detection with Human Data

Given our model of a rigid collision, each of the three detectors will eventually detect the collision, since the magnitude of the applied force will continue to increase over time towards infinity after

onset of the event. As such, for this evaluation we focus on the increase in the magnitude of the force from the onset of the haptic event until its detection, and the false positive rate.

This increase in the magnitude of the force is the excess force applied to the mechanism before the detector reports that a haptic event occurred. False positives correspond to the detector reporting a haptic event for a collision-free trial.

We assume that an ideal rigid collision is equally likely to occur across all configurations of all operations of all mechanisms. Given these assumptions, we can evaluate all three detectors using a database that only contains captured haptic data associated with successful, collision-free operation of various mechanisms.

Let us assume that the door makes an ideal collision with a fixed rigid obstacle at a configuration x. For a given value of f_x^{thresh}, the excess force before a collision is detected will be $f_x^{thresh} - \hat{f}$, where $\hat{f}$ is the estimated tangential force at configuration x for one collision-free trial of pulling open the mechanism door.

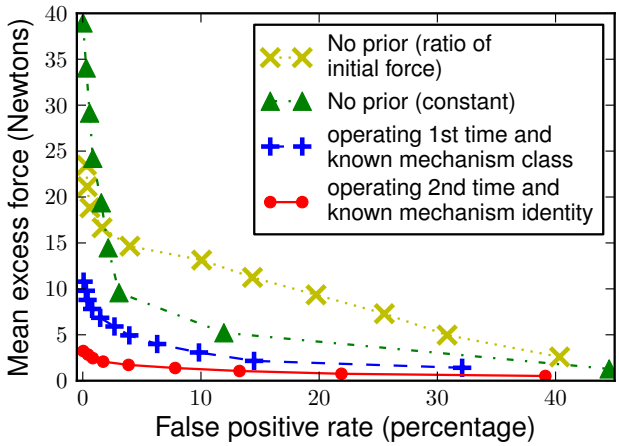
We compute the average value of this excess force over all the configurations of all the trials of all the mechanisms to obtain the mean excess force before an anomalous force is detected. This gives us the y coordinate of a point in the plot of fig. 92 on page 135. The x axis is the false positive percentage, which is the percentage of the configurations for which $f_x^{thresh} < \hat{f}$. This is a false positive as we know that our database contains only collision-free trials.

Figure 92 shows the performance of the three detectors for different values of n, r, and c on 148 trials from 28 different rotary mechanisms belonging to 5 different classes. Each point in the plot represents a different value of f_x^{thresh}, obtained by changing the value of the parameters n, r, and c for the different methods of detecting anomalous forces. For all detectors, increasing the free parameter (n, r, or c) decreases the false positive rate, but increases the unnecessary force applied prior to detection.

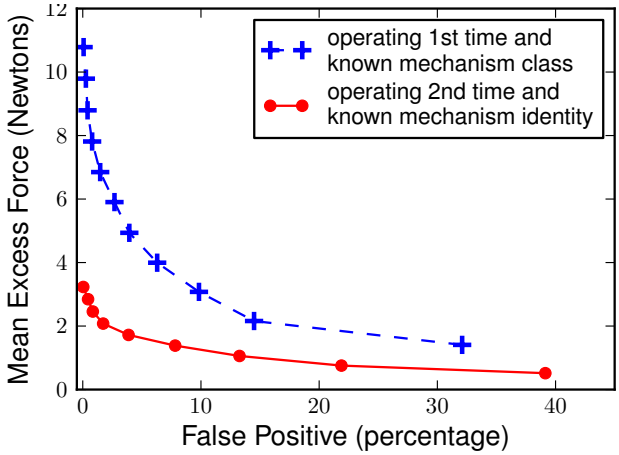
Figure 92a shows that using captured haptic data can reduce the excess force for a given false positive rate. It shows that knowledge of the mechanism class enables the detection of an anomalous force with a lower excess force (for any false positive rate, the blue curve is below the green and yellow curves). Additionally, it shows that even a single trial with the specific mechanism decreases the excess force further, as evidenced by the fact that the red curve is below the blue curve for all false positive rates.

Figure 92b shows the performance of the detector of eq. (64) on page 132 for different values of n in more detail. The blue curve is when the robot operates a mechanism for the first time, and only knows its class. The red curve is when the robot is operating a mechanism for the second time and

consequently has one haptic interaction vector, defined in section 7.4.1 on page 123, in addition to knowledge of the mechanism's class.



(a) Using haptic data results in lower excess force for the same false positive rate.



(b) One previous trial with the specific mechanism results in lower excess force for the same false positive rate.

Figure 92: *Graphs illustrating how the different anomalous force detection methods trade off the mean excess force before reporting a collision (y axis), with the false positive rate or percentage of times that a collision free trail is incorrectly reported as being in collision (x axis). These results use the dataset of human trials.*

7.8 Experiments with Two Robots

In this section we report results on haptic event detection using two robots, Cody and a Willow Garage PR2, briefly described in section 1.2 on page 3. We explain our method for online estimation of the mechanism state and tangential forces, show how it would affect the detection of anomalous forces, and present results from trials on the two robots. Figure 87 on page 126 shows the two

mobile manipulators and fig. 84 on page 122 shows how we mounted a hook end effector to them.

7.8.1 Online Estimation

To detect haptic events using the methods of section 7.7.1 on page 132, a robot needs to generate online estimates of the state of the mechanism and the tangential component of the force while operating the mechanism.

Our method for online estimation first estimates the radius of the trajectory of the handle, r, and the location of the axis of rotation, (c_x, c_y), which it then uses to compute an estimate of the mechanism state and tangential force. We will denote (r, c_x, c_y) with θ.

We assume that a perception algorithm gives the robot an initial estimate of the radius of the trajectory of the handle, r_p. As an example, the perception algorithm could compute r_p based on the estimated width of the door and the location of the handle [182]. In addition, the robot estimates the pose of the tip of the hook using forward kinematics while operating the mechanism. This gives it an estimate of the trajectory of the handle, $((x_1, y_1), (x_2, y_2), ...(x_n, y_n))$ which we denote as T_n. The number of points in the trajectory of the handle, n, increases with time.

Given r_p and T_n we compute a maximum likelihood estimate [26] of θ as

$$\hat{\theta} = \underset{\theta}{\operatorname{argmax}}\, P(T_n, r_p|\theta), \tag{65}$$

We assume that the observed trajectory, T_n, and the perception algorithm's estimate of the radius, r_p, are conditionally independent given θ. So,

$$P(T_n, r_p|\theta) = P(T_n|\theta)P(r_p|\theta). \tag{66}$$

Next, we assume that the perception algorithm's estimate of the radius is normally distributed around the true radius of the mechanism, with a variance of σ_r^2. So,

$$P(r_p|\theta) = \frac{1}{\sqrt{2\pi\sigma_r^2}} e^{-\frac{(r_p - r)^2}{2\sigma_r^2}}. \tag{67}$$

We compute $P(T|\theta)$ using the assumptions detailed in [206], which include assuming that the measurements of the points along the trajectory of the handle, (x_i, y_i), are conditionally independent given θ, and have a Gaussian error with a variance of σ_{pos}^2. Equation (65) then simplifies to

$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}} \frac{(r - r_p)^2}{\sigma_r^2} + \sum_{i=1}^{n} \frac{\left(r - \sqrt{(c_x - x_i)^2 + (c_y - y_i)^2}\right)^2}{\sigma_{pos}^2}, \tag{68}$$

which we optimize using the implementation of the BFGS algorithm from SciPy [102]. We then use $\hat{\theta}$ to compute the current state of the mechanism. For our tests, we set $\sigma_r = 10cm$, and $\sigma_{pos} = 1cm$.

We believe these values are conservative given the capabilities of current perception algorithms, the resolution of the joint encoders, and our use of a hook with a layer of rubber to pull on the door handle.

We now look at the performance of the anomalous force detection methods described in section 7.7.1 on page 132 with online mechanism state estimation. We collected the trajectory of the handle and the forces applied to the mechanism for five collision-free trials of the PR2 and Cody opening two office cabinets, and five trials of Cody opening a refrigerator.

We simulated multiple trials of online estimation of the mechanism state, given uncertainty in the initial estimate of radius of the door as follows: for each of the 15 trials with the robots, we computed multiple values of r_p in eq. (68) by sampling from a Gaussian with mean equal to the true radius of the mechanism and a standard deviation of 10cm. We then used $\hat{\theta}$ from eq. (68) to generate online estimates of the state of the mechanism and the component of the force tangential to the trajectory of the handle. As a result, we simulated multiple haptic interaction vectors with online state estimation and a noisy initial estimate of the radius r_p.

Figure 93 on the following page shows the trade-off between the false positive rate and the sensitivity of the anomalous force detection, analogous to the results from section 7.7.4 on page 133. Figure 93 shows that errors in the estimates of the mechanism configuration (due to uncertainty about the radius) result in poorer performance when the robot operates a mechanism for the first time (higher force on average before collision is detected). We expect an improvement in performance if the robot uses methods for kinematic estimation that yield accurate initial estimates of the radius of the door and the state of the mechanism, such as in [182, 206, 205].

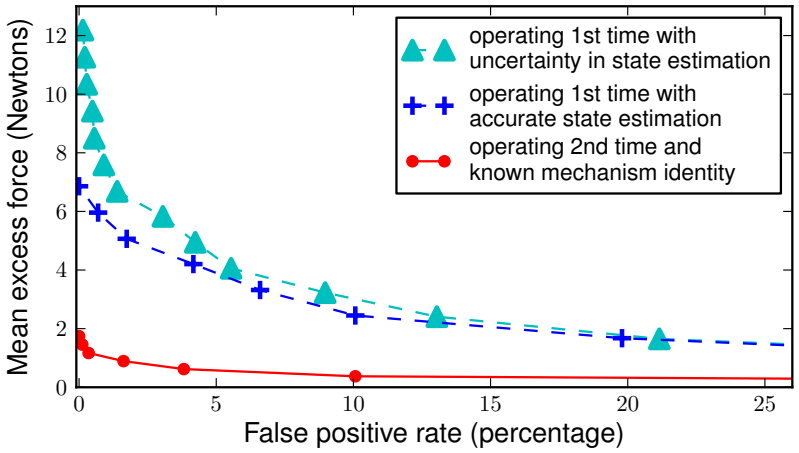


Figure 93: *Effect of accuracy of kinematic state estimation on performance of anomalous force detection using data captured with Cody and the PR2.*

7.8.2 Collision Detection

We performed six trials with the two robots, shown in fig. 94 on page 139. We either placed an obstacle in front of the mechanism, or the door was locked. We processed data collected from these trials off-line for anomalous force detection using the detector based on our probabilistic model of applied forces (eq. (64) on page 132). We set n in eq. (64) as the least value that results in zero false positives on our dataset of human trials. Further, we assumed that the robot has an accurate estimate of the radius of the mechanism (as opposed to being drawn from a Gaussian around the true radius as described in section 7.8.1 on page 136).

Table 10 on page 140 shows the time between the onset of the collision and when it would be detected, and the excess force applied to the handle for the six trials of fig. 94.

In fig. 94 on the following page, the red curve is the tangential force measured during the trial. If the trial were collision-free, we would expect the red curve to be close to the solid blue line, which is the mean of the collision-free tangential forces applied to open that specific mechanism. The dashed green and blue lines are the minimum forces at which the anomaly detection method would report a collision if the robot were operating the mechanism for the first time (with knowledge of the mechanism class), or the second time, respectively. Only the dashed curves determine when an anomalous force would be detected. We have shown the solid blue curve for visualization only.

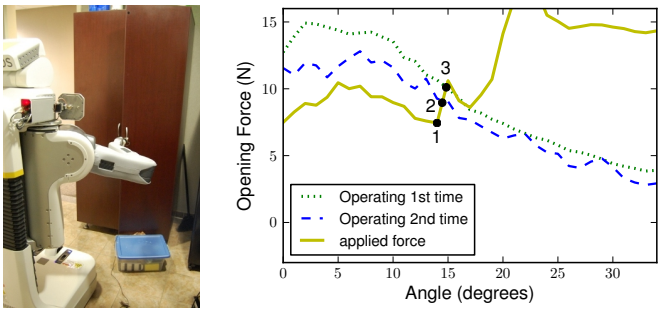
The three numbered circles (1 – 3) in fig. 94 represent the onset of the collision event, the point at which the robot would have detected a collision if it were operating the mechanism for the second time, and the point at which it would have detected a collision if it were operating the mechanism for the first time.

For trials 3 and 4, the robot has a slightly lower force threshold for detecting anomalous conditions when it is operating the refrigerator for the first time than when it is operating it the second time. We believe that this is because the database of human trials currently has data from a small number (four) of refrigerators resulting in the mean and variance not being representative of the mechanism class.

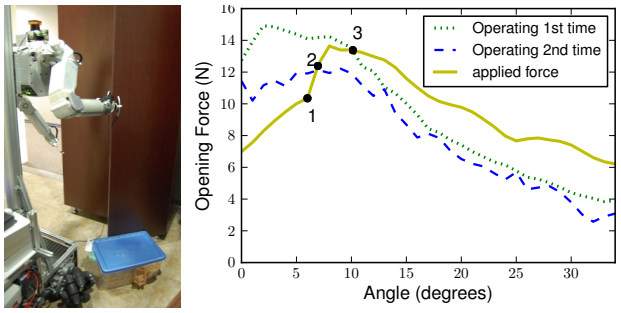
7.9 Discussion

7.9.1 Broader Implications

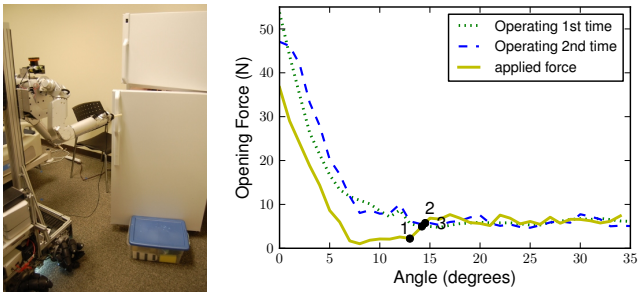
Machine intelligence has benefitted greatly from large collections of sensory data. Web-based databases of user-generated content, such as videos from YouTube, images from Flickr, and 3D models from Google 3D Warehouse, have begun to support the development of robots and enabling technology [113, 121, 115, 119, 216]. More generally, research has shown that large datasets can



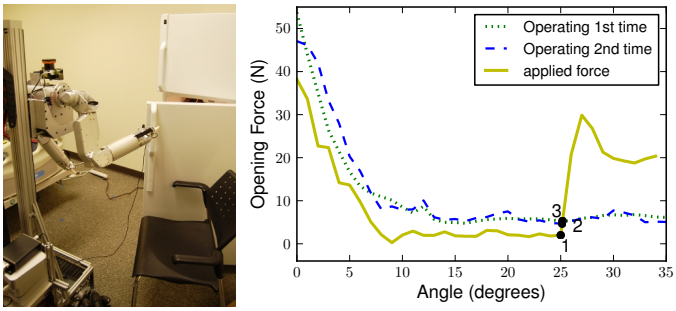
(a) Trial 1 – Cabinet door collides with a box.



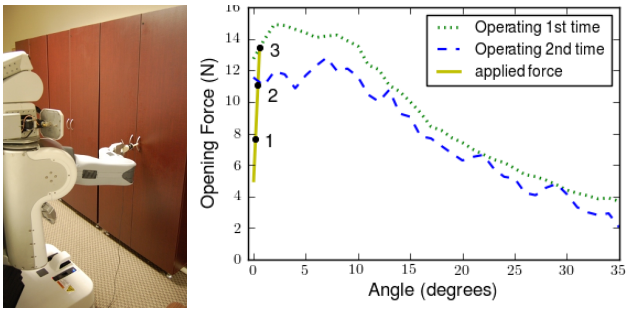
(b) Trial 2 – Cabinet door collides with a box.



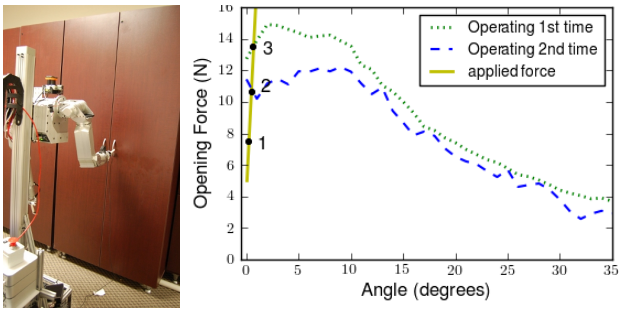
(c) Trial 3 – Refrigerator door collides with a box.



(d) Trial 4 – Refrigerator door collides with a chair.



(e) Trial 5 – Locked cabinet door.



(f) Trial 6 – Locked cabinet door.

Figure 94: *Six trials with two different robots to test the anomaly detection methods. Each set of images shows a picture of the mechanism and the object with which the door collides, and a graph that gives details of the trial. In the graph, the red curve is the tangential force measured during the trial, the solid blue curve is the mean of the collision-free tangential forces applied to operate the mechanism. The dashed green and dashed blue curves are the minimum forces at which the robot would report a collision if it were operating the mechanism for the first or the second time, respectively. The three numbered circles indicate the onset of the collision, and the points when the two anomaly detection methods would have reported a collision.*

lead to performance gains and make computationally simple methods effective [213, 73].

Our results suggest that humans and robots have the potential to share haptic data to improve robot manipulation. Additionally, by storing associated contextual information, such as where the interactions occurred and the appearance of manipulated objects, robots could anticipate haptic interactions.

Table 10: *Performance of the Anomaly Detection Methods on trials with Cody and the PR2.*

	Detection time (secs) Open door the		Excess Force (N) Open door the	
	1^{st} time	2^{nd} time	1^{st} time	2^{nd} time
Trial 1	1.1	0.3	2.6	1.9
Trial 2	2.4	0.6	3.1	2.1
Trial 3	1.5	1.8	2.6	3.5
Trial 4	0.5	0.5	3.7	3.7
Trial 5	25.5	6.1	5.7	3.4
Trial 6	1.4	0.7	6.2	3.4
Mean (std)	5.4 (9.0)	1.7 (2.0)	4.0 (1.4)	3.0 (0.7)

In general, this type of common sense knowledge would help robots behave more intelligently. In the future, robots might use these data in numerous ways, including selecting better postures prior to manipulation, detecting when mechanisms are in need of repair, and anticipating when a human will require assistance. Moreover, these data could be used by humans to rationally design robots with the kinematic and force capabilities necessary to perform real-world tasks.

Enabling humans to easily capture the haptic interactions, such as with a wearable system, could potentially accelerate the accumulation of this type of data. With motion capture capabilities and sensors continuing to improve in quality and lower in cost, there is the potential for the robotics community to accumulate large datasets in a practical manner. Likewise, robots in the field could potentially record their haptic interactions and upload them to an online database to produce a continuously evolving source of haptic knowledge for various manipulation tasks.

7.9.2 Limitations and Future Work

7.9.2.1 More Tasks, More Data, More Robots

Scaling up our approach to more manipulation tasks, more mechanisms and trials, and more robots is an important area for future inquiry. In this chapter, we have presented results with real-world data for one manipulation task: slowly and smoothly pulling open doors. In future work, we would like to use the methods presented in this paper for more tasks. These could include twisting door knobs, pushing buttons on appliances, and inserting a cell phone charger into a wall socket. It could also include tasks relevant to activities of daily living (ADLs) such as bed baths [112], shaving, grooming, and manipulating a person's body.

Data-driven object-centric models of haptic interactions for these tasks may enable robots to efficiently detect anomalous conditions without excessive force. For example, a robot may be able to haptically detect that it is attempting to insert the incorrect key or that the door is not completely

shut. Likewise, a robot may stop inserting a USB plug, flip it, and retry.

For our current work, we identified the relevant applied forces and the relevant mechanism state, and found a useful low-dimensional model by using our task knowledge, modeling the kinematics, and experimenting with various models. For example, we found that a quasi-static model was sufficient for low speed door opening and we did not need to include the angular velocity or angular acceleration of the door in the relevant mechanism state. Similar models might work for other tasks with 1 degree-of-freedom kinematics, such as twisting a door knob or turning a key. More generally, methods to automate aspects of the modeling process would be desirable. Machine learning might be able to autonomously discover appropriate low-dimensional representations.

We have tested our methods on a dataset of forces from 148 trials of human operators opening 28 doors in 6 homes and one office, and 21 trials from two robots opening three doors in one office. Our results are promising, and suggest the potential for scaling up to more trials, mechanisms, and robots, but actually doing so remains an open area for inquiry. Ideally, the robotics community will begin to collect large scale datasets of forces from everyday activities to facilitate progress, much like the computer vision community.

7.9.2.2 Haptic Data from Different Sensors

In this work, humans and two robots used a hook instrumented with the same six-axis force-torque sensor at the base while pulling open doors. Additionally, we restricted contact between the hook and the door to be at the handle. In general, other sensors, such as joint torque sensors and tactile sensors, might be used to record the forces.

We have shown that the component of the force tangential to the trajectory of the door handle depends on the mechanism and not on the control method used to open the door. However, different sensors will have varying accuracy and noise levels. For example, using joint torque sensing to estimate the force at the end effector will be affected by the dynamics of the arm and friction and flexibility in the joints. In this paper, we do not discuss methods for combining data with varying accuracy and amounts of noise into a common database of haptic interactions.

Robot-centric models, discussed in section 7.2.2.1 on page 119, do not provide a direct way for different robots to share information but make it easier for a robot to use new sensors by representing the haptic interaction directly in terms of the robot's state and sensors. Our method will require additional effort to transform the measurements from different sensors into an object-centric representation, but offers the potential for multiple robots to share the transformed data and models.

7.9.2.3 Haptic Event Detection

Recent work on haptic event detection has demonstrated the use of high frequency information to detect haptic events such as collisions while placing an object on a table [179]. Our models make use of relatively low-frequency haptic sensing. Incorporating high-frequency features and other features into the models might be beneficial. Likewise, weakening the assumption that the forces are independent given the mechanism configuration could lead to better detection of haptic events.

7.10 Conclusion

We have demonstrated that humans and robots can capture haptic data while performing a manipulation task in a form that they can directly share in spite of variations in their bodies and control methods. We recorded the relevant forces, the points of application of the forces, and mechanism state while humans and two different robots pulled open doors at low speeds. We then represented the haptic interactions in an object-centric representation that could be shared by different robots. We demonstrated that these data could be used to haptically identify mechanism classes and specific mechanisms, and build data-driven object-centric models that enable robots to detect collisions between the door and the environment faster and with a lower excess force.

More generally, we have presented a method for building probabilistic data-driven task-specific object-centric models of haptic interactions. These models can be shared by different robots for improved manipulation performance. We have used pulling open doors, a important task for service robots, as an example to demonstrate our method.

CHAPTER VIII

CONCLUSION

This thesis takes a first step towards a new foundation for robot manipulation with an approach that encourages robots to make contact with the world. Our approach is different from the current paradigm of using non-contact perception to avoid contact with the world except at end effector locations.

Robot arms have been used very successfully for tasks that are repeatable and need to be performed with fast precise motions, for example welding on an automotive assembly line. As a result, performance is often synonymous with precision, speed, and payload. A majority of the commercially available robot arms have been designed to be stiff and position controlled to permit fast, precise, and repeatable motions. These characteristics have the side effect of increasing the undesriable consequences of contact (such as large forces resulting in damage to the robot or the environment), increasing the importance of avoiding contact.

We believe that placing a priority on precise motions, non-contact perception, and avoiding contact with the world has resulted robots that dramatically underperform humans and animals in manipulation in unstructured environments. Inspired by biology, our approach has been to use compliant actuation at the joints and whole-arm tactile sensing to reduce the cost of contact between the robot and its environment and enable robots to manipulate while making contact across all surfaces of their arms.

8.1 Possible Directions for Future Research

8.1.1 Whole-Arm Manipulation (Reprise)

In this thesis, we have presented a controller for reaching to a goal location in cluttered environments. The ability of the arm to push and bend movable and compliant objects is currently a useful by-product of the goal directed motion and our assumption that contact force below a threshold has no associated penalty. Future work could include using all the surfaces of the arm to apply and sense forces while performing manipulation tasks such as grasping and pushing aside large objects, manipulating a person's body, and haptically searching an environment.

Our assumption that contact can occur on all surfaces of a robot arm shares similarities with the idea of whole-arm manipulation [184]. Initial research on whole-arm manipulation had focused on the electro-mechanical design and low-level control for torque-controlled robot arms, for example

[185, 214, 54]. Now, robot arms with compliant actuation at the joints are commercially available, such as the WAM Arm from Barrett Technology, arms from Meka Robotics and the Personal Robot 2 (PR2) from Willow Garage. New companies such as Rethink Robotics and Redwood Robotics may make compliant arms accessible to more researchers by producing inexpensive compliant arms.

The initial research did not address controlling the contact forces or performing tasks with robots. Additionally, as discussed in chapter 2 on page 7, research on control for robots that make contact with the environment at multiple locations across their arms has been less common. In this thesis, we have presented a controller for reaching to goal locations in cluttered environments with the possibility of multiple contacts across the entire surface of the arm.

Initial research on whole-arm manipulation had also identified some potential benefits of tactile sensing (e.g. [67]). There has been a long history of research on tactile sensing technologies [36, 38], and there are some recent examples of robots with large surfaces of their arms covered with tactile sensors (e.g., [190, 87]). In this thesis, we have presented the design of inexpensive and easy to fabricate tactile sensors that we have used to cover large surfaces of two compliant robot arms.

Given the current status of compliant arms and whole-arm tactile sensing, we believe that there is a renewed opportunity for research on control whole-arm manipulation.

8.1.2 Higher Level Decisions and Haptic Mapping

The controller that we have developed only uses the most recent tactile sensor feedback to attempt to move the arm towards the goal while keeping the contact forces low. We currently discard all previous measurements of the contact forces and contact locations. Adding memory to our system could enable the robot to make higher-level decisions, such as repositioning the mobile base or reaching from a different position to maneuver around an object that is fixed and rigid and can not be pushed aside.

One approach for a robot to maintain a memory of the haptic feedback would be to use techniques from simultaneous localization and mapping to build a haptic map of the environment. This map could include estimates of the mechanical properties of objects in different regions, such as the stiffness, and whether an object is movable or fixed. A haptic map could also be augmented with data from non-contact sensors.

For the task of reaching to a goal location, a haptic map could allow a robot to reason about good starting configurations for the mobile base and arm. A haptic map could also enable a robot to anticipate contact before it is detected using the tactile sensors, enabling the robot to increase the speed of operation while still regulating contact forces. More generally, knowledge of the mechanical properties of the environment may be useful to plan how to rearrange clutter and haptically

search for an object within a volume.

8.1.3 Statistical Models of Forces

In chapter 7 on page 116, we have presented data-driven methods to inform robots about the forces that they are likely to encounter when performing specific tasks. In the context of door opening we have shown that data-driven models can be used to haptically recognize specific doors, haptically recognize classes of door and detect anomalous forces.

A possible direction for future research could be to build statistical models different environments from whole-arm tactile sensor feedback. For example, the statistics of forces while manipulating in contact with objects with different mechanical properties, such as branches and leaves in foliage, stones in rubble, and pipes and cables in machinery, could be different. Data-driven models of the forces could be used to haptically find objects (such as a box hidden in foliage), detect anomalous conditions, or haptically recognize different environments and enable the selection of strategies and control parameters that are better matched to a particular environment.

8.2 Specific Contributions of Individual Chapters

We have made the following contributions towards our borader goal of enabling robots to manipulate effectively with haptic feedback and compliant actuation at the joints:

- In chapter 2 on page 7, we have presented our approach to robot manipulation, which emphasizes the benefits of making contact with the world across all the surfaces of a manipulator. We have developed a controller that enables robots with whole-arm tactile sensing and compliant actuation at the joints to reach to locations in high clutter while regulating contact forces. At every time step, the controller generates a mechanical model of the arm interacting with its environment and uses this to move the arm towards the goal location while keeping the predicted contact forces low.

- In chapter 3 on page 24, we have evaluated our controller in software simulation and on a real robot. In our experiments, these robots pushed aside movable objects, deformed compliant objects, and perceived the world through contact. Our tests included a real robot reaching into a cinder block and simulated foliage using a novel tactile sensor array on its forearm. In simulation, we have also shown that our controller performs better with whole-arm tactile sensing than with a force-torque sensor at each link, and that the relative value of whole-arm tactile sensing increases with higher clutter.

- In chapter 4 on page 46, we have presented our design for inexpensive and easy to fabricate

tactile sensor arrays using conductive and resistive fabric. We have covered the end effector, wrist joints, and forearm of Cody, and large surfaces of a PR2 arm with tactile sensor arrays with low spatial resolution. With an informal test, we have shown that our controller of chapter 2 and this sensor have the potential to enable robots to manipulate in close proximity to, and in contact with humans. We have also presented a single test of manipulating an object on a shelf with multiple contacts between the robot arm and the environment.

- In chapter 5 on page 63, we have presented a set of feedback controllers that enable a mobile manipulator to reliably and autonomously approach and open doors and drawers for which only the location and orientation of the handle have been provided. We have empirically demonstrated that our controllers are robust with respect to variations in the mechanism, the pose of the base, the stiffness of the arm, the friction between the door handle and the end effector, and the way the handle was hooked. We have also demonstrated that our controllers can coordinate the movement of the robot's omnidirectional base and compliant arm while pulling open a door or drawer.

- In chapter 6 on page 99, we have presented a custom force and motion capture system that we built to capture the forces and trajectories while pulling open doors and drawers. We have shown that the forces while opening these seemingly simple everyday devices have complex structure that is hard to explain with mechanical models but can be captured and characterized using data-driven methods.

- In chapter 7 on page 116, we have presented data-driven methods to inform robots about the forces that they are likely to encounter when performing specific tasks. In the context of door opening, we have demonstrated that data-driven models can be used to haptically recognize specific doors, haptically recognize classes of door (e.g., refrigerator vs. kitchen cabinet), and better detect haptic events (e.g., a locked door or collision), even when opening a specific door for the first time. We have also shown that two distinct robots can use forces captured from people opening doors to better detect haptic events. This illustrates the potential for databases of task-specific forces to be used to improve robot manipulation.

These contributions constitute a first step towards our longer-term goal of a new foundation for mobile manipulation. We would like to enable robot to seek out and make contact with the world using all surfaces of their arms and have common sense about the forces involved in different manipulation tasks.

APPENDIX A

CODE, DATA, VIDEOS, AND HARDWARE DESIGNS

A.1 Reaching in Clutter with Whole-Arm Tactile Sensing

Python code, videos, and hardware designs for the end effector and hardware-in-the-loop simulation testbed are available at: `www.hsi.gatech.edu/hrl/reach-clutter-journal.shtml`

A.2 Fabric Tactile Sensors

Video of Cody reaching through a pipe:
`http://www.youtube.com/watch?v=reG2OUOM8Iw`
Video of teleoperating a PR2 to manipulate on a shelf:
`http://www.youtube.com/watch?v=p6e4MfBdH3Y`
Video of Henry Evans teleoperating a PR2 around his body:
`http://www.youtube.com/watch?v=JRF73DwK7jI`
Python code, videos, and hardware designs for the fabric tactile sensor available at:
`www.hsi.gatech.edu/hrl/project_fabric_tactile_sensor.shtml`

A.3 Door Opening

- Python code for the feedback controllers, videos and a CAD model of the hook end effector are available at:
  `www.hsi.gatech.edu/hrl/epc-icra10.shtml`
  `www.hsi.gatech.edu/hrl/epc-humanoids09.shtml`

- Hardware designs and Python code for the Hokuyo mounted on a Robotis servo, as shown in fig. 44 on page 65, are available at:
  `http://www.hsi.gatech.edu/hrl-wiki/index.php/Tilting_Hokuyo`

A.4 Capturing the Forces and Trajectories that Open Doors and Drawers

A video of the data capture process, and code and hardware designs for our custom force and motion capture system can be found at: `www.hsi.gatech.edu/hrl/mechanics-biorob10.shtml`

A.5 Improving Robot Manipulation with Data-Driven Object-Centric Models of Everyday Forces

Python code and data associated with chapter 7 are available at:

`www.hsi.gatech.edu/hrl/improve-forces-journal.shtml`

REFERENCES

[1] http://www.woodbin.com/ref/furniture/cabinets.htm. 107

[2] http://www.hometips.com/buying-guides/kitchen-cabinets-sizes.html. 107

[3] "http://wildshores.blogspot.com/2009/06/wild-monkeys-at-breakfast-in-admi
html." 9

[4] "http://www.ethantw.com/noodling.html, http://en.wikipedia.org/wiki/Noodling." 9

[5] "http://www.nativeamerica.com/research.html." 9

[6] "Americans with Disabilities Act Accessibility Guidelines (ADAAG)." http://www.access-board.gov/adaag/html/adaag.htm, 1992. 112

[7] "Open source computer vision library." Intel Corporation, 2001. 101

[8] "Arduino Mega 2560 R3," http://arduino.cc/it/Main/ArduinoBoardMega2560. 50

[9] "McDavid compression sleeveless t-shirt," http://www.amazon.com/McDavid-compression-sleeveless-t-shirt/dp/B0069UBQN0. 49

[10] "EeonTex Resistive Fabric," http://www.eeonyx.com/datasheets.php?type=eeontex. 49

[11] "LessEMF Stretch Conductive Fabric," http://www.lessemf.com/fabric.html. 49

[12] ABBEEL, P., COATES, A., and NG, A., "Autonomous helicopter aerobatics through apprenticeship learning," *The International Journal of Robotics Research*, vol. 29, no. 3, pp. 1608–1639, 2010. 13

[13] ALBU-SCHAFFER, A., OTT, C., FRESE, U., and HIRZINGER, G., "Cartesian impedance control of redundant robots: Recent results with the dlr-light-weight-arms," in *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on*, vol. 3, pp. 3704–3709, IEEE, 2003. 16

[14] ALEXANDER, R., "Three uses for springs in legged locomotion," *The International Journal of Robotics Research*, vol. 9, no. 2, pp. 53–61, 1990. 1, 10

[15] ALIREZAEI, H., NAGAKUBO, A., and KUNIYOSHI, Y., "A highly stretchable tactile distribution sensor for smooth surfaced humanoids," in *Humanoid Robots, 2007 7th IEEE-RAS International Conference on*, pp. 167–173, Ieee, 2007. 47

[16] ALIREZAEI, H., NAGAKUBO, A., and KUNIYOSHI, Y., "A tactile distribution sensor which enables stable measurement under high and dynamic stretch," in *3D User Interfaces, 2009. 3DUI 2009. IEEE Symposium on*, pp. 87–93, IEEE, 2009. 47

[17] ANGERILLI, M., FRISOLI, A., SALSEDO, F., MARCHESCHI, S., and BERGAMASCO, M., "Haptic simulation of an automotive manual gearshift," in *ROMAN*, 2001. 101, 118

[18] ARKIN, R., "Motor schema-based mobile robot navigation," *International Journal of Robotics Research*, vol. 8, no. 4, pp. 92–112, 1989. 89

[19] BECKER, J., BERSCH, C., PANGERCIC, D., PITZER, B., RÜHR, T., SANKARAN, B., STURM, J., STACHNISS, C., BEETZ, M., and BURGARD, W., "The pr2 workshop-mobile manipulation of kitchen containers," in *IROS Workshop on Results, Challenges and Lessons Learned in Advancing Robots with a Common Platform*, 2011. 120

[20] BELLINGHAM, J., RICHARDS, A., and HOW, J., "Receding horizon control of autonomous aerial vehicles," in *American Control Conference*, 2002. 13

[21] BIANCHI, L., "Mechanotransduction: Touch and feel at the molecular level as modeled in caenorhabditis elegans," *Molecular Neurobiology*, vol. 36, no. 3, pp. 254–271, 2007. 1, 9

[22] BICCHI, A., "Force distribution in multiple whole-limb manipulation," in *IEEE International Conference on Robotics and Automation*, 1993. 12

[23] BICCHI, A. and KUMAR, V., "Robotic grasping and contact: A review," in *IEEE International Conference on Robotics and Automation.*, 2000. 12

[24] BICCHI, A., SALISBURY, J., and BROCK, D., "Contact sensing from force measurements," *The International Journal of Robotics Research*, vol. 12, no. 3, pp. 249–262, 1993. 10

[25] BIERBAUM, A., ASFOUR, T., and DILLMANN, R., "Dynamic potential fields for dexterous tactile exploration," *Human Centered Robot Systems*, pp. 23–31, 2009. 13

[26] BISHOP, C., *Pattern recognition and machine learning*. Springer New York, 2006. 130, 131, 136

[27] BOYD, S. and VANDENBERGHE, L., *Convex optimization*. Cambridge Univ Pr, 2004. 19

[28] BROOKS, R., ARYANANDA, L., EDSINGER, A., FITZPATRICK, P., KEMP, C., O'REILLY, U., TORRES-JARA, E., VARSHAVSKAYA, P., and WEBER, J., "Sensing and Manipulating Built-for-Human Environments," *International Journal of Humanoid Robotics*, vol. 1, no. 1, pp. 1–28, 2004. 64

[29] BRUYNINCKX, H. and DE SCHUTTER, J., "Specification of force-controlled actions in the "Task Frame Formalism": A Synthesis," *Robotics and Automation, IEEE Transactions on*, vol. 12, no. 4, pp. 581–589, 1996. 81, 88

[30] BUERGER, S. and HOGAN, N., "Complementary stability and loop shaping for improved human–robot interaction," *IEEE Transactions on Robotics*, vol. 23, no. 2, pp. 232–244, 2007. 10

[31] BYL, K. and TEDRAKE, R., "Approximate optimal control of the compass gait on rough terrain," in *IEEE International Conference on Robotics and Automation*, 2008. 45

[32] CATANIA, K., "A nose that looks like a hand and acts like an eye: the unusual mechanosensory system of the star-nosed mole," *Journal of Comparative Physiology A: Neuroethology, Sensory, Neural, and Behavioral Physiology*, vol. 185, no. 4, pp. 367–372, 1999. 9

[33] CHITTA, S., STURM, J., PICCOLI, M., and BURGARD, W., "Tactile sensing for mobile manipulation," *Robotics, IEEE Transactions on*, no. 99, pp. 1–11, 2011. 13

[34] CHITTA, S., COHEN, B., and LIKHACHEV, M., "Planning for autonomous door opening with a mobile manipulator," in *ICRA*, 2010. 64, 120

[35] CHOI, Y. S., ANDERSON, C. D., GLASS, J. D., and KEMP, C. C., "Laser pointers and a touch screen: Intuitive interfaces to an autonomous mobile robot for the motor impaired," in *ACM SIGACCESS*, 2008. 66, 86

[36] CUTKOSKY, M. R., HOWE, R. D., and PROVANCHER, W. R., *Force and Tactile Sensing*, ch. 19. Springer Handbook of Robotics, 2008. 1, 46, 47, 144

[37] CUTKOSKY, M. R. and ULMEN, J. V., "Dynamic Tactile Sensing," in *The Human Hand: A Source of Inspiration for Robotic Hands* (SANTOS, V. and BALASUBRAMANIAN, R., eds.), ch. 24, Springer Berlin Heidelberg, 2012. 28, 46

[38] DAHIYA, R., METTA, G., VALLE, M., and SANDINI, G., "Tactile sensingfrom humans to humanoids," *IEEE Transactions on Robotics*, vol. 26, no. 1, pp. 1–20, 2010. 46, 144

[39] DE LUCA, A., ALBU-SCHAFFER, A., HADDADIN, S., and HIRZINGER, G., "Collision detection and safe reaction with the dlr-iii lightweight manipulator arm," in *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pp. 1623–1630, Ieee, 2006. 12

[40] DE LUCA, A. and MATTONE, R., "An adapt-and-detect actuator fdi scheme for robot manipulators," in *ICRA*, 2004. 12, 119

[41] DE SCHUTTER, J., BRUYNINCKX, H., DUTRÉ, S., DE GEETER, J., KATUPITIYA, J., DEMEY, S., and LEFEBVRE, T., "Estimating first-order geometric parameters and monitoring contact transitions during force-controlled compliant motion," *The International Journal of Robotics Research*, vol. 18, no. 12, pp. 1161–1184, 1999. 10

[42] DIANKOV, R., "Checkerboard Detector ROS Package." `http://www.ros.org/wiki/checkerboard_detector`, 2009. 104

[43] DIANKOV, R. and KUFFNER, J., "Openrave: A planning architecture for autonomous robotics," *Robotics Institute, Pittsburgh, PA, Tech. Rep. CMU-RI-TR-08-34*, 2008. 27, 31

[44] DIANKOV, R., SRINIVASA, S., FERGUSON, D., and KUFFNER, J., "Manipulation planning with caging grasps," in *Humanoids*, 2008. 64, 120

[45] DIXON, W., WALKER, I., DAWSON, D., and HARTRANFT, J., "Fault detection for robot manipulators with parametric uncertainty: a prediction-error-based approach," *Robotics and Automation, IEEE Transactions on*, vol. 16, no. 6, pp. 689–699, 2000. 119

[46] DOGAR, M. and SRINIVASA, S., "A framework for push-grasping in clutter," *Robotics: Science and Systems*, 2011. 12

[47] DOGAR, M., HEMRAJANI, V., LEEDS, D., KANE, B., and SRINIVASA, S., "Proprioceptive Localization for Mobile Manipulators," tech. rep., Carnegie Mellon University, 2010. 10

[48] DOLLAR, A. M. and HOWE, R. D., "Towards grasping in unstructured environments: Grasper compliance and configuraton optimization," *Advanced Robotics*, 2005. 112

[49] DOMINY, N., "Fruits, fingers, and fermentation: the sensory cues available to foraging primates," *Integrative and Comparative Biology*, vol. 44, no. 4, pp. 295–303, 2004. 9

[50] DUCHAINE, V., BOUCHARD, S., and GOSSELIN, C., "Computationally efficient predictive robot control," *Mechatronics, IEEE/ASME Transactions on*, vol. 12, no. 5, pp. 570–578, 2007. 13

[51] DUPONT, P., SCHULTEIS, C., MILLMAN, P., and HOWE, R., "Automatic identification of environment haptic properties," *Presence*, vol. 8, no. 4, pp. 394–411, 1999. 101, 118

[52] DUPONT, P. and YAMAJAKO, S., "Jamming and wedging in constrained rigid-body dynamics," in *Robotics and Automation, 1994. Proceedings., 1994 IEEE International Conference on*, pp. 2349–2354, IEEE, 1994. 10

[53] EBERMAN, B. and SALISBURY, J., "Determination of manipulator contact information from joint torque measurements," in *Experimental Robotics I*, 1990. 10

[54] EBERMAN, B., "Whole-arm manipulation: kinematics and control," Master's thesis, MIT, 1989. 10, 144

[55] EBERST, C., ANDERSSON, M., and CHRISTENSEN, H., "Vision-based door-traversal for autonomous mobile robots," in *Proceedings of the 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2000. 64

[56] EDSINGER, A. and KEMP, C. C., "Human-robot interaction for cooperative manipulation: Handing objects to one another," in *Proceedings of the 16th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, 2007. 16

[57] EDSINGER, A. and KEMP, C. C., "Two arms are better than one: A behavior-based control system for assistive bimanual manipulation," in *Proceedings of the 13th International Conference on Advanced Robotics (ICAR)*, 2007. 16

[58] EREZ, T., TASSA, Y., and TODOROV, E., "Infinite-horizon model predictive control for periodic tasks with contacts," in *Robotics: Science and Systems (RSS)*, 2011. 13

[59] ESCANDE, A. and KHEDDAR, A., "Contact planning for acyclic motion with tasks constraints," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2009. 12

[60] FEATHERSTONE, R. and ORIN, D. E., *Chapter 2: Dynamics, Handbook of Robotics, Siciliano, Bruno; Khatib, Oussama (Eds.)*. Springer, 2008. 17

[61] FRANK, B., STACHNISS, C., ABDO, N., and BURGARD, W., "Using gaussian process regression for efficient motion planning in environments with deformable objects," in *Workshops at the Twenty-Fifth AAAI Conference on Artificial Intelligence*, 2011. 13

[62] FROM, P., GRAVDAHL, J., LILLEHAGEN, T., and ABBEEL, P., "Motion planning and control of robotic manipulators on seaborne platforms," *Control engineering practice*, vol. 19, no. 8, pp. 809–819, 2011. 13

[63] GARCIA, C., PRETT, D., and MORARI, M., "Model predictive control: theory and practicea survey," *Automatica*, vol. 25, no. 3, pp. 335–348, 1989. 13

[64] GARCIA, M., CHATTERJEE, A., RUINA, A., and COLEMAN, M., "The simplest walking model: Stability, complexity, and scaling," *Journal of Biomechanical Engineering*, vol. 120, p. 281, 1998. 13

[65] GLASER, S., "Teleop Controllers ROS Package." Willow Garage, Robot Operating System, 2010. 123

[66] GOODMAN, M. B., *WormBook*, ch. Mechanosensation. 2006. 1, 9

[67] GORDON, S. and TOWNSEND, W., "Integration of tactile force and joint torque information in a whole-arm manipulator," in *IEEE International Conference on Robotics and Automation*, 1989. 144

[68] GORGES, N., NAVARRO, S., GOGER, D., and WORN, H., "Haptic object recognition using passive joints and haptic key features," in *ICRA*, 2010. 119

[69] GROSSO, M., QUACH, R., and BADLER, N., "Anthropometry for computer animated human figures," *State-of-the Art in Computer Animation*, p. 83, 1989. 25

[70] GU, X. and BALLARD, D., "An equilibrium point based model unifying movement control in humanoids," in *RSS*, 2006. 16

[71] HADDADIN, S., ALBU-SCHAFFER, A., DE LUCA, A., and HIRZINGER, G., "Collision detection and reaction: A contribution to safe physical human-robot interaction," in *IROS*, 2008. 12, 119

[72] HADDADIN, S., ALBU-SCHAFFER, A., HADDADIN, F., ROSMANN, J., and HIRZINGER, G., "Study on soft-tissue injury in robotics," *IEEE Robotics & Automation Magazine*, vol. 18, no. 4, pp. 20–34, 2011. 119

[73] HALEVY, A., NORVIG, P., and PEREIRA, F., "The unreasonable effectiveness of data," *Intelligent Systems, IEEE*, vol. 24, no. 2, pp. 8–12, 2009. 2, 140

[74] HAUSER, K., BRETL, T., and LATOMBE, J., "Non-gaited humanoid locomotion planning," in *Humanoids*, 2005. 12

[75] HERSCH, M. and BILLARD, A., "A biologically-inspired controller for reaching movements," in *IEEE/RAS-EMBS International Conference on Biomedical Robotics and Biomechatronics (BIOROB)*, 2006. 11

[76] HOGAN, N., "Adaptive control of mechanical impedance by coactivation of antagonist muscles," *Automatic Control, IEEE Transactions on*, vol. 29, no. 8, pp. 681–690, 1984. 1, 10

[77] HOGAN, N., "The mechanics of multi-joint posture and movement control," *Biological cybernetics*, vol. 52, no. 5, pp. 315–331, 1985. 63

[78] HOGAN, N., "On the stability of manipulators performing contact tasks," *Robotics and Automation, IEEE Journal of*, vol. 4, no. 6, pp. 677–686, 1988. 16, 22

[79] HOGAN, N. and BUERGER, S., *Impedance and Interaction Control*, ch. 19. Robotics and Automation Handbook, 2005. 15, 16, 22, 66, 79, 87

[80] HOSHI, T. and SHINODA, H., "A sensitive skin based on touch-area-evaluating tactile elements," in *Haptic Interfaces for Virtual Environment and Teleoperator Systems, 2006 14th Symposium on*, pp. 89–94, IEEE, 2006. 47, 48

[81] HSIAO, K., CHITTA, S., CIOCARLIE, M., and JONES, E., "Contact-reactive grasping of objects with partial shape information," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2010. 11

[82] HSIAO, K., KAELBLING, L., and LOZANO-PÉREZ, T., "Task-driven tactile exploration," *Robotics: Science and Systems*, 2010. 12

[83] HSIAO, K. and LOZANO-PEREZ, T., "Imitation learning of whole-body grasps," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2006. 12

[84] INABA, M., HOSHINO, Y., NAGASAKA, K., NINOMIYA, T., KAGAMI, S., and INOUE, H., "A full-body tactile sensor suit using electrically conductive fabric and strings," in *Intelligent Robots and Systems' 96, IROS 96, Proceedings of the 1996 IEEE/RSJ International Conference on*, vol. 2, pp. 450–457, IEEE, 1996. 47

[85] IVALDI, S., FUMAGALLI, M., NORI, F., BAGLIETTO, M., METTA, G., and SANDINI, G., "Approximate optimal control for reaching and trajectory planning in a humanoid robot," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2010. 13

[86] IWANIUK, A. and WHISHAW, I., "How skilled are the skilled limb movements of the raccoon (procyon lotor)?," *Behavioural brain research*, vol. 99, no. 1, pp. 35–44, 1999. 9

[87] IWATA, H. and SUGANO, S., "Design of human symbiotic robot twendy-one," in *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pp. 580–586, IEEE, 2009. 144

[88] J. PRATT, M. CHEW, P. D. and PRATT, G., "Virtual Model Control: An intuitive approach for bipedal locomotion," *International Journal of Robotics Research*, vol. 20, no. 2, pp. 129–143, 2001. 10, 13

[89] JAIN, A. and KEMP, C. C., "Improving Robot Manipulation with Data-Driven Object-Centric Models of Everyday Forces," *Under review*. 116

[90] JAIN, A. and KEMP, C. C., "Behaviors for robust door opening and doorway traversal with a force-sensing mobile manipulator," in *RSS Workshop on Robot Manipulation: Intelligence in Human Environments*, 2008. 74

[91] JAIN, A. and KEMP, C. C., "Behavior-based door opening with equilibrium point control," in *RSS Workshop: Mobile Manipulation in Human Environments*, 2009. 16, 63, 120

[92] JAIN, A. and KEMP, C. C., "Pulling Open Novel Doors and Drawers with Equilibrium Point Control," in *Humanoids*, 2009. 16, 26, 63

[93] JAIN, A. and KEMP, C. C., "EL-E: An Assistive Mobile Manipulator that Autonomously Fetches Objects from Flat Surfaces," *Autonomous Robots*, 2010. 11, 112

[94] JAIN, A. and KEMP, C. C., "Pulling Open Doors and Drawers: Coordinating an Omni-directional Base and a Compliant Arm with Equilibrium Point Control," in *ICRA*, 2010. 16, 26, 63, 99, 120, 123

[95] JAIN, A., KILLPACK, M. D., EDSINGER, A., and KEMP, C. C., "Reaching in Clutter with Whole-Arm Tactile Sensing," *Under review*. 7

[96] JAIN, A., NGUYEN, H., RATH, M., OKERMAN, J., and KEMP, C. C., "The Complex Structure of Simple Devices: A Survey of Trajectories and Forces that Open Doors and Drawers," in *Proceedings of the IEEE RAS/EMBS International Conference on Biomedical Robotics and Biomechatronics (BIOROB)*, 2010. 99, 117, 121

[97] JOHANSSON, R., "How is grasping modified by somatosensory input," *Motor control: Concepts and issues*, pp. 331–355, 1991. 1

[98] JOHANSSON, R. and FLANAGAN, J., "Coding and use of tactile signals from the fingertips in object manipulation tasks," *Nature Reviews Neuroscience*, vol. 10, no. 5, pp. 345–359, 2009. 1

[99] JOHANSSON, R. and WESTLING, G., "Roles of glabrous skin receptors and sensorimotor memory in automatic control of precision grip when lifting rougher or more slippery objects," *Experimental Brain Research*, vol. 56, no. 3, pp. 550–564, 1984. 1

[100] JOHNSON, K. and JOHNSON, K., *Normal Contact of Elastic Solids: Hertz Theory*, ch. 4. Contact Mechanics, 1987. 18

[101] JOHNSSON, M. and BALKENIUS, C., "Neural network models of haptic shape perception," *Robotics and Autonomous Systems*, vol. 55, no. 9, pp. 720–727, 2007. 119

[102] JONES, E., OLIPHANT, T., PETERSON, P., and OTHERS, "SciPy: Open source scientific tools for Python," *http://www. scipy. org*, 2001. 82, 131, 137

[103] KALAKRISHNAN, M., RIGHETTI, L., PASTOR, P., and SCHAAL, S., "Learning force control policies for compliant manipulation," in *IROS*, 2011. 120

[104] KANEKO, M. and TANIE, K., "Contact point detection for grasping an unknown object using self-posture changeability," *Robotics and Automation, IEEE Transactions on*, vol. 10, no. 3, pp. 355–367, 1994. 10

[105] KAO, I., LYNCH, K., and BURDICK, J. W., *Contact Modeling and Manipulation*, ch. 27. Springer Handbook of Robotics, 2008. 18

[106] KATZ, D. and BROCK, O., "Extracting planar kinematic models using interactive perception," *Unifying Perspectives in Computational and Robot Vision*, 2008. 99

[107] KAVRAKI, L. E. and LAVALLE, S. M., *Chapter 5: Motion Planning, Handbook of Robotics, Siciliano, Bruno; Khatib, Oussama (Eds.)*. Springer, 2008. 11

[108] KEMP, C. C., FITZPATRICK, P., HIRUKAWA, H., YOKOI, K., HARADA, K., and MATSUMOTO, Y., *Chapter 56: Humanoids, Handbook of Robotics, Siciliano, Bruno; Khatib, Oussama (Eds.)*. Springer, 2008. 111

[109] KERPA, O., WEISS, K., and WORN, H., "Development of a flexible tactile sensor system for a humanoid robot," in *Intelligent Robots and Systems, 2003.(IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, vol. 1, pp. 1–6, IEEE, 2003. 46

[110] KHATIB, O., "A unified approach for motion and force control of robot manipulators: The operational space formulation," *Robotics and Automation, IEEE Journal of*, vol. 3, no. 1, pp. 43–53, 1987. 12

[111] KILLPACK, M., DEYLE, T., ANDERSON, C., and KEMP, C., "Visual odometry and control for an omnidirectional mobile robot with a downward-facing camera," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010. 3, 26, 86, 87

[112] KING, C. H., CHEN, T. L., and KEMP, C. C., "Towards an assistive robot that autonomously performs bed baths for patient hygiene," in *Intelligent Robots and Systems, 2010 (IROS 2010), IEEE/RSJ International Conference on*, pp. 319–324, 2010. 141

[113] KLANK, U., ZIA, M., and BEETZ, M., "3d model selection from an internet database for robotic vision," in *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pp. 2406–2411, IEEE, 2009. 2, 140

[114] KLINGBEIL, E., SAXENA, A., and NG, A. Y., "Learning to open new doors," in *RSS Workshop on Robot Manipulation: Intelligence in Human Environments*, 2008. 64, 120

[115] KOLLAR, T. and ROY, N., "Utilizing object-object and object-scene context when planning to find things," in *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pp. 2168–2173, IEEE, 2009. 2, 140

[116] KORMUSHEV, P., CALINON, S., and CALDWELL, D., "Imitation learning of positional and force skills demonstrated via kinesthetic teaching and haptic input," *Advanced Robotics*, vol. 25, no. 5, pp. 581–603, 2011. 120

[117] KRAGIC, D. and CHRISTENSEN, H., "A Framework for Visual Servoing Tasks," *Intelligent Autonomous Systems*, 2000. 64

[118] KROSHKO, D. L., "Openopt framework. `http://openopt.org`," 2011. 19

[119] KUFFNER, J., "Cloud Enabled Humanoid Robots," *Humanoids: What's next? Applications, Challenges and Perspectives Workshop*, 2010. 2, 140

[120] KULCHENKO, P. and TODOROV, E., "First-exit model predictive control of fast discontinuous dynamics: Application to ball bouncing," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2011. 13

[121] LAI, K. and FOX, D., "3D laser scan classification using web data and domain adaptation," in *Proc. of Robotics: Science and Systems (RSS)*, 2009. 2, 140

[122] LAVALLE, S. and KUFFNER, J., "Randomized kinodynamic planning," *The International Journal of Robotics Research*, vol. 20, no. 5, p. 378, 2001. 11

[123] LEDERMAN, S. and KLATZKY, R., "Haptic perception: A tutorial," *Attention, Perception, & Psychophysics*, vol. 71, no. 7, pp. 1439–1459, 2009. 1, 9

[124] LEEPER, A., HSIAO, K., CIOCARLIE, M., TAKAYAMA, L., and GOSSOW, D., "Strategies for human-in-the-loop robotic grasping," in *ACM/IEEE international conference on Human Robot Interaction – To Appear*, 2012. 11

[125] LEGAGNE, S., KHEDDAR, A., and YOSHIDA, E., "Generation of Optimal Dynamic Multi-Contact Motions : Application to Humanoid Robots," *IEEE Transactions on Robotics – under review*, 2011. 12

[126] LOZANO-PEREZ, T., "A simple motion-planning algorithm for general robot manipulators," *Robotics and Automation, IEEE Journal of*, vol. 3, no. 3, pp. 224–238, 1987. 11

[127] LUMPKIN, E. A., MARSHALL, K. L., and NELSON, A. M., "The cell biology of touch," *Journal of Cell Biology*, vol. 191, pp. 237–248, October 2010. 1, 9

[128] LUTSCHER, E., LAWITZKY, M., CHENG, G., and HIRCHE, S., "A control strategy for operating unknown constrained mechanisms," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pp. 819–824, IEEE. 64

[129] MACLEAN, K., "The haptic camera: A technique for characterizing and playing back haptic properties of real environments," in *Proceedings of ASME Dynamic Systems and Control Division*, 1996. 101, 118

[130] MALADEN, R., DING, Y., UMBANHOWAR, P., KAMOR, A., and GOLDMAN, D., "Biophysically inspired development of a sand-swimming robot," *Robotics: Science and Systems (RSS)*, 2010. 13

[131] MANCHESTER, I., METTIN, U., IIDA, F., and TEDRAKE, R., "Stable dynamic walking over uneven terrain," *The International Journal of Robotics Research*, vol. 30, no. 3, pp. 265–279, 2011. 13

[132] MASON, M., "Compliance and force control for computer-controlled manipulators," *IEEE Trans on Systems, Man, and Cybernetics*, 1981. 104

[133] MASON, M., "Compliance and force control for computer controlled manipulators," *IEEE Transactions on Systems, Man and Cybernetics*. 81, 88

[134] MASON, M., *Mechanics of robotic manipulation.* 2001. 10

[135] MASON, M., RODRIGUEZ, A., SRINIVASA, S., and VAZQUEZ, A., "Autonomous manipulation with a general-purpose simple hand," *International Journal of Robotics Research*, vol. 31, no. 5, pp. 688–703, 2012. 12

[136] MATHEUS, K. and DOLLAR, A. M., "Benchmarking Grasping and Manipulation: Properties of the Objects of Daily Living," in *IROS*, 2010. 112, 118

[137] MAZZINI, F., KETTLER, D., DUBOWSKY, S., and GUERRERO, J., "Tactile robotic mapping of unknown surfaces: an application to oil well exploration," in *Robotic and Sensors Environments, 2009. ROSE 2009. IEEE International Workshop on*, pp. 80–85, IEEE, 2009. 13

[138] MCKENNA, J., ANHALT, D., BRONSON, F., BROWN, H., SCHWERIN, M., SHAMMAS, E., and CHOSET, H., "Toroidal skin drive for snake robot locomotion," in *International Conference on Robotics and Automation*, 2008. 13

[139] METTA, G., NATALE, L., NORI, F., and SANDINI, G., "Force control and reaching movements on the icub humanoid robot," *International Symposium on Robotics Research*, 2011. 11

[140] MIGLIORE, S., *The Role of Passive Joint Stiffness and Active Knee Control in Robotic Leg Swinging: Applications to Dynamic Walking*. PhD thesis, Georgia Institute of Technology, 2009. 16

[141] MIGLIORE, S., BROWN, E., and DEWEERTH, S., "Biologically inspired joint stiffness control," in *IEEE International Conference on Robotics and Automation*, 2005. 1, 10

[142] MORARI, M. and LEE, J. H., "Model predictive control: past, present and future," *Computers & Chemical Engineering*, vol. 23, no. 4-5, pp. 667–682, 1999. 14

[143] MORINAGA, S. and KOSUGE, K., "Collision detection system for manipulator based on adaptive impedance control law," in *ICRA*, 2003. 119

[144] MURRAY, R., LI, Z., and SASTRY, S., *A mathematical introduction to robotic manipulation*. CRC, 1994. 30

[145] NAGATANI, K. and YUTA, S., "An experiment on opening-door-behavior by an autonomous mobile robot with a manipulator," *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 45–50, 1995. 64

[146] NATALE, L. and TORRES-JARA, E., "A sensitive approach to grasping," in *International Workshop on Epigenetic Robotics*, 2006. 11

[147] NIEMEYER, G. and SLOTINE, J., "A simple strategy for opening an unknown door," in *1997 IEEE International Conference on Robotics and Automation, 1997. Proceedings.*, vol. 2, 1997. 64

[148] OTT, C., BAEUML, B., BORST, C., and HIRZINGER, G., "Autonomous opening of a door with a mobile manipulator: A case study," *IFAC Symposium on Intelligent Autonomous Vehicles*, 2007. 64

[149] PAI, D., LANG, J., LLOYD, J., and WOODHAM, R., "Acme, a telerobotic active measurement facility," *Experimental Robotics VI*, pp. 391–400, 2000. 118

[150] PAPAKOSTAS, T., LIMA, J., and LOWE, M., "A large area force sensor for smart skin applications," in *Sensors, 2002. Proceedings of IEEE*, vol. 2, pp. 1620–1624, IEEE, 2002. 47

[151] PARK, J. and KHATIB, O., "Robot multiple contact control," *Robotica*, vol. 26, no. 5, 2008. 12

[152] PASTOR, P., KALAKRISHNAN, M., CHITTA, S., THEODOROU, E., and SCHAAL, S., "Skill learning and task outcome prediction for manipulation," in *ICRA*, 2011. 119

[153] PASTOR, P., RIGHETTI, L., KALAKRISHNAN, M., and SCHAAL, S., "Online movement adaptation based on previous sensor experiences," in *IROS*, 2011. 11

[154] PATIL, S., VAN DEN BERG, J., and ALTEROVITZ, R., "Motion planning under uncertainty in highly deformable environments," in *Robotics: Science and Systems (RSS)*, 2011. 13

[155] PERNER-WILSON, H. and DANFORTH, I., "rSkin – Open Source Robot Skin," `http://www.instructables.com/id/rSkin-Open-Source-Robot-Skin/`. 47

[156] PETERSSON, L., AUSTIN, D., and KRAGIC, D., "High-level control of a mobile manipulator for door opening," *International Conference on Intelligent Robots and Systems*, 2000. 64

[157] PETROVSKAYA, A. and KHATIB, O., "Global localization of objects via touch," *Robotics, IEEE Transactions on*, vol. 27, no. 3, pp. 569–585, 2011. 13

[158] PETROVSKAYA, A. and NG, A., "Probabilistic Mobile Manipulation in Dynamic Environments, with Application to Opening Doors," *International joint conference on artificial intelligence (IJCAI07), Hyderabad*, 2007. 64

[159] PETROVSKAYA, A., PARK, J., and KHATIB, O., "Probabilistic Estimation of Whole Body Contacts for Multi-Contact Robot Control," *IEEE International Conference on Robotics and Automation*, 2007. 12

[160] PETROVSKYA, A., KHATIB, O., THRUN, S., and NG, A., "Bayesian estimation for autonomous object manipulation based on tactile sensors," *Proc. of ICRA*, 2006. 64

[161] PEZZEMENTI, Z., PLAKU, E., REYDA, C., and HAGER, G., "Tactile-object recognition from appearance information," *Robotics, IEEE Transactions on*, vol. 27, no. 3, pp. 473–487, 2011. 13

[162] PLATT JR, R., FAGG, A., and GRUPEN, R., "Extending fingertip grasping to whole body grasping," in *IEEE International Conference on Robotics and Automation*, 2003. 12

[163] PRATS, M., SANZ, P., and DEL POBIL, A., "Reliable non-prehensile door opening through the combination of vision, tactile and force feedback," *Autonomous Robots*, vol. 29, no. 2, pp. 201–218, 2010. 64

[164] PRATS, M., SANZ, P., DEL POBIL, A., MARTÍNEZ, E., and MARÍN, R., "Towards multipurpose autonomous manipulation with the UJI service robot," *Robotica*, vol. 25, no. 02, pp. 245–256, 2007. 64

[165] PRATS, M., WIELAND, S., ASFOUR, T., DEL POBIL, A., and DILLMANN, R., "Compliant interaction in household environments by the Armar-III humanoid robot," in *Humanoids*, 2008. 64, 81, 88

[166] PRATT, G., "Low impedance walking robots 1," *Integrative and Comparative Biology*, vol. 42, no. 1, pp. 174–181, 2002. 10, 13

[167] PRATT, G. and WILLIAMSON, M., "Series elastic actuators," in *IROS*, 1995. 3, 10, 26

[168] PRESCOTT, T., PEARSON, M., FOX, C., EVANS, M., MITCHINSON, B., ANDERSON, S., and PIPE, T., "Towards biomimetic vibrissal tactile sensing for robot exploration, navigation, and object recognition in hazardous environments," in *RISE'10, Fourth internationnal workshop on Robotics for Risky Interventions and Environmental Surveillance-Maintenance*, 2010. 13

[169] QUIGLEY, M., BATRA, S., GOULD, S., KLINGBEIL, E., LE, Q., WELLMAN, A., and NG, A. Y., "High-Accuracy 3D Sensing for Mobile Manipulation: Improving Object Detection and Door Opening," in *ICRA*, 2009. 64

[170] QUIGLEY, M., GERKEY, B., CONLEY, K., FAUST, J., FOOTE, T., LEIBS, J., ERIC BERGER, R. W., and NG, A., "ROS: An Open-Source Robot Operating System," in *ICRA Open-Source Software workshop*, 2009. 25, 58, 101

[171] RAIBERT, M., BLANKESPOOR, K., NELSON, G., PLAYTER, R., and OTHERS, "Bigdog, the rough–terrain quadruped robot," in *Proceedings of the 17th World Congress*, 2008. 13

[172] RAIBERT, M. and CRAIG, J., "Hybrid position/force control of manipulators," *Journal of Dynamic Systems, Measurement, and Control*, vol. 102, no. 127, pp. 126–133, 1981. 12

[173] RAPACKI, E., NIEZRECKI, C., and YANCO, H., "An underactuated gripper to unlatch door knobs and handles," in *Technologies for Practical Robot Applications, 2009. TePRA 2009. IEEE International Conference on*, pp. 135–140, IEEE. 64

[174] REDMOND, B., AINA, R., GORTI, T., and HANNAFORD, B., "Haptic characteristics of some activities of daily living," in *Haptics Symposium, 2010 IEEE*, pp. 71–76, IEEE, 2010. 118

[175] RHEE, C., CHUNG, W., KIM, M., SHIM, Y., and LEE, H., "Door opening control using the multi-fingered robotic hand for the indoor service robot," *Proceedings. ICRA'04. 2004 IEEE International Conference on Robotics and Automation*, 2004. 64

[176] RODRIGUEZ, A., BOURNE, D., MASON, M., ROSSANO, G., and WANG, J., "Failure detection in assembly: Force signature analysis," in *IEEE Conference on Automation Science and Engineering (CASE)*, 2010. 119

[177] RODRIGUEZ, S., LIEN, J., and AMATO, N., "Planning motion in completely deformable environments," in *IEEE International Conference on Robotics and Automation*, 2006. 13

[178] ROMANO, J. and KUCHENBECKER, K., "Creating realistic virtual textures from contact acceleration data," *Haptics, IEEE Transactions on*, no. 99, 2011. 118

[179] ROMANO, J., HSIAO, K., NIEMEYER, G., CHITTA, S., and KUCHENBECKER, K., "Human-inspired robotic grasp control with tactile sensing," *Robotics, IEEE Transactions on*, no. 99, pp. 1–13, 2011. 11, 142

[180] RÜHR, T., STURM, J., PANGERCIC, D., BEETZ, M., and CREMERS, D., "A generalized framework for opening doors and drawers in kitchen environments," in *ICRA*, 2012. 120

[181] RUSU, R. B., MEEUSSEN, W., CHITTA, S., and BEETZ, M., "Laser-based perception for door and handle identification," in *Proceedings of International Conference on Advanced Robotics*, 2009. 67

[182] RUSU, R., MARTON, Z., BLODOW, N., DOLHA, M., and BEETZ, M., "Towards 3D Point cloud based object maps for household environments," *Robotics and Autonomous Systems*, vol. 56, no. 11, pp. 927–941, 2008. 67, 136, 137

[183] SALISBURY, J., J., "Interpretation of contact geometries from force measurements," in *ICRA*, vol. 1, mar 1984. 10

[184] SALISBURY, K., "Whole arm manipulation," in *Proceedings of the 4th international symposium on Robotics Research*, pp. 183–189, MIT Press, 1988. 143

[185] SALISBURY, K., TOWNSEND, W., EBRMAN, B., and DIPIETRO, D., "Preliminary design of a whole-arm manipulation system (wams)," in *Robotics and Automation, 1988. Proceedings., 1988 IEEE International Conference on*, pp. 254–260, IEEE, 1988. 144

[186] SARANLI, U., BUEHLER, M., and KODITSCHEK, D., "Rhex: A simple and highly mobile hexapod robot," *The International Journal of Robotics Research*, vol. 20, no. 7, pp. 616–631, 2001. 13

[187] SATAVA, R. and JONES, S., "Virtual environments for medical training and education," *Presence: Teleoperators and Virtual Environments*, vol. 6, no. 2, pp. 139–146, 1997. 101

[188] SAXENA, A., DRIEMEYER, J., and NG, A., "Robotic Grasping of Novel Objects using Vision," *The International Journal of Robotics Research*, vol. 27, no. 2, p. 157, 2008. 11

[189] SCHMID, A., GORGES, N., GÖGER, D., and WÖRN, H., "Opening a door with a humanoid robot using multi-sensory tactile feedback," *International Conference on Robotics and Automation*, 2008. 64

[190] SCHMITZ, A., MAIOLINO, P., MAGGIALI, M., NATALE, L., CANNATA, G., and METTA, G., "Methods and technologies for the implementation of large-scale robot tactile sensors," *IEEE Transactions on Robotics*, no. 99, pp. 1–12, 2011. 46, 48, 53, 144

[191] SCHNEIDER, A., STURM, J., STACHNISS, C., REISERT, M., BURKHARDT, H., and BURGARD, W., "Object identification with tactile sensors using bag-of-features," in *IROS*, IEEE, 2009. 119

[192] SENTIS, L. and KHATIB, O., "Synthesis of whole-body behaviors through hierarchical control of behavioral primitives," *International Journal of Humanoid Robotics*, 2005. 12

[193] SENTIS, L., PARK, J., and KHATIB, O., "Compliant Control of Multicontact and Center-of-Mass Behaviors in Humanoid Robots," *IEEE Transactions on Robotics*, vol. 26, no. 3, 2010. 12, 16

[194] SHADMEHR, R., "Control of equilibrium position and stiffness through postural modules," *Journal of motor behavior*, vol. 25, no. 3, pp. 228–241, 1993. 26, 83

[195] SHIMOJO, M., NAMIKI, A., ISHIKAWA, M., MAKINO, R., and MABUCHI, K., "A tactile sensor sheet using pressure conductive rubber with electrical-wires stitched method," *Sensors Journal, IEEE*, vol. 4, no. 5, pp. 589–596, 2004. 47

[196] SINAPOV, J., BERGQUIST, T., SCHENCK, C., OHIRI, U., GRIFFITH, S., and STOYTCHEV, A., "Interactive object recognition using proprioceptive and auditory feedback," *The International Journal of Robotics Research*, vol. 30, no. 10, pp. 1250–1262, 2011. 119

[197] SMABY, N., JOHANSON, M., BAKER, B., KENNEY, D., MURRAY, W., and HENTZ, V., "Identification of key pinch forces required to complete functional tasks," *Journal of Rehabilitation Research and Development*, vol. 41, no. 2, pp. 215–224, 2004. 118

[198] SMITH, R. and OTHERS, "Open dynamics engine `http://www.ode.org`," 2011. 25

[199] SMITS, R. and OTHERS, "Kinematics and dynamics library `http://www.orocos.org/kdl`." 65

[200] SRINIVASA, S., FERGUSON, D. HELFRICH, C., BERENSON, D., COLLET, A., DIANKOV, R., GALLAGHER, G., HOLLINGER, G., KUFFNER, J., and VANDEWEGHE, M., "Herb: A Home Exploring Robotic Butler," *Autonomous Robots*, 2009. 11

[201] STILMAN, M., NISHIWAKI, K., and KAGAMI, S., "Learning object models for whole body manipulation," *Humanoids*, 2007. 99

[202] STILMAN, M., NISHIWAKI, K., KAGAMI, S., and KUFFNER, J., "Planning and executing navigation among movable obstacles," *Advanced Robotics*, vol. 21, no. 14, pp. 1617–1634, 2007. 12

[203] STILMAN, M., SCHAMBUREK, J., KUFFNER, J., and ASFOUR, T., "Manipulation planning among movable obstacles," in *IEEE Int. Conf. on Robotics and Automation*, 2007. 11, 12

[204] STULP, F., KRESSE, I., MALDONADO, A., RUIZ, F., FEDRIZZI, A., and BEETZ, M., "Compact models of human reaching motions for robotic control in everyday manipulation tasks," in *IEEE International Conference on Development and Learning*, 2009. 11

[205] STURM, J., JAIN, A., STACHNISS, C., KEMP, C., and BURGARD, W., "Operating Articulated Objects Based on Experience," in *IROS*, 2010. 120, 137

[206] STURM, J., PRADEEP, V., STACHNISS, C., PLAGEMANN, C., KONOLIGE, K., and BURGARD, W., "Learning kinematic models for articulated objects," in *Proc. of the Int. Conf. on Artificial Intelligence (IJCAI)*, 2009. 99, 104, 136, 137

[207] STURM, J., JAIN, A., STACHNISS, C., KEMP, C. C., and BURGARD, W., "Operating Articulated Objects Based on Experience," in *IROS*, 2010. 113

[208] SUKHOY, V., GEORGIEV, V., WEGTER, T., SWEIDAN, R., and STOYTCHEV, A., "Learning to slide a magnetic card through a card reader," in *ICRA*, 2012. 119

[209] . 76

[210] TAKAMUKU, S., FUKUDA, A., and HOSODA, K., "Repetitive grasping with anthropomorphic skin-covered hand enables robust haptic recognition," in *IROS*, 2008. 119

[211] TAWIL, D., RYE, D., and VELONAKI, M., "Improved eit drive patterns for a robotics sensitive skin," in *Proceeding of Australasian Conference on Robotics and Automation (ACRA)*, pp. 2–4, 2009. 47

[212] THIBODEAU, B., DEEGAN, P., and GRUPEN, R., "Static analysis of contact forces with a mobile manipulator," in *ICRA*, 2006. 114

[213] TORRALBA, A., FERGUS, R., and FREEMAN, W., "80 million tiny images: A large data set for nonparametric object and scene recognition," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 30, no. 11, pp. 1958 –1970, 2008. 2, 140

[214] TOWNSEND, W., *The effect of transmission design on the performance of force-controlled manipulators*. PhD thesis, Doctoral thesis, Massachusetts Institute of Technology, 1988. 144

[215] W. MEEUSSEN ET AL., "Autonomous door opening and plugging in with a personal robot," in *ICRA*, 2010. 64, 112, 120

[216] WAIBEL, M., BEETZ, M., CIVERA, J., D'ANDREA, R., ELFRING, J., GALVEZ-LOPEZ, D., HAUSSERMANN, K., JANSSEN, R., MONTIEL, J., PERZYLO, A., and OTHERS, "Roboearth," *Robotics & Automation Magazine, IEEE*, vol. 18, no. 2, pp. 69–82, 2011. 2, 140

[217] WEIR, D., PEHKIN, M., COLGATE, J., BUTTOLO, P., RANKIN, J., and JOHNSTON, M., "The haptic profile:capturing the feel of switches," in *HAPTICS*, 2004. 101, 118

[218] WHITNEY, D., "Quasi-static assembly of compliantly supported rigid parts," *Journal of Dynamic Systems, Measurement, and Control*, vol. 104, p. 65, 1982. 10

[219] WHITNEY, D. and NEVINS, J., "What is the remote center compliance (rcc) and what can it do," in *Proceedings of the 9th International Symposium on Industrial Robots*, pp. 135–152, 1979. 10

[220] WIEBER, P., "Trajectory free linear model predictive control for stable walking in the presence of strong perturbations," in *IEEE-RAS International Conference on Humanoid Robots*, 2006. 13

[221] WIELAND, S., GONZALEZ-AGUIRRE, D., VAHRENKAMP, N., ASFOUR, T., and DILLMANN, R., "Combining force and visual feedback for physical interaction tasks in humanoid robots," in *Humanoids*, 2009. 64, 120

[222] WILLIAMSON, M., "Postural primitives: Interactive behavior for a humanoid robot arm," in *Proceedings of the Fourth International Conference on Simulation of Adaptive Behavior*, 1996. 16

[223] WILLIAMSON, M., *Robot arm control exploiting natural dynamics*. PhD thesis, Massachusetts Institute of Technology, 1999. 16

[224] WISTE, T., DALLEY, S., VAROL, H., and GOLDFARB, M., "Design of a multigrasp transradial prosthesis," *Journal of Medical Devices*, vol. 5, 2011. 118

[225] WYROBEK, K., BERGER, E., VAN DER LOOS, H., and SALISBURY, J., "Towards a personal robotics development platform: Rationale and design of an intrinsically safe personal robot," in *ICRA*, 2008. 4, 112